Sintaxis y Semántica de los lenguajes

Expresiones regulares y equivalencias entre representaciones de lenguajes regulares.

2022

Facultad Regional Delta,
Universidad Tecnológica Nacional

Prof. Juan Miguel Santos

Expresiones regulares.

Equivalencias entre las representaciones de los lenguajes regulares.

- Pasaje de autómata finito a expresión regular (T).
- Pasaje de expresión regular a autómata finito (T).
- Pasaje de gramática regular a AFND-λ
- Pasaje de un AFD a una GR. Algoritmo (T).
- Pasaje de gramática regular a AFND. Algoritmo.

Expresiones regulares:

Es un formalismo para expresar un lenguaje regular.

Sea Σ un alfabeto, las expresiones regulares sobre Σ y los conjuntos que ellas denotan (conjuntos regulares) son definidos recursivamente como sigue:

1) ϕ es una e.r. y denota el conjunto vacío ϕ.

2) λ es una e.r. y denota el conjunto $\{\lambda\}$

3) Por cada **a** $\in \Sigma$, **a** es una e.r. y denota el conjunto {**a**} con sólo la cadena **a**.

4) Si **r** y **s** son e.r. denotando los lenguajes R y S, entonces

   r / **s** es una e.r. y denota $R \cup S$

   rs es una e.r. y denota RoS

   r* es una e.r. y denota el conjunto R*

Nota:

Orden de precedencia : ((0(1*))/0) puede escribirse como 01*/0

Abreviaturas: rr* es equivalente a r^+

Ejemplo:

Sea el alfabeto Σ = { 0, 1 }:

-00 es una e.r. que denota el lenguaje { 00 },

-(0/1)* es una e.r. que denota el lenguaje de todas las cadenas con 0s y 1s,

-(0/1)*00(0/1)* es una e.r. que denota todas las cadenas de 0s y 1s con al menos dos 0s consecutivos dentro de ellas,

-(1/10)* es una e.r. que denota el lenguaje cuyas cadenas son 0s y 1s empezando con 1 y sin dos 0s consecutivos,

-(0/1)*011 es una e.r. que denota el lenguaje cuyas cadenas son 0s y 1s que terminan siempre con 011.

Sea el alfabeto Σ = {0, 1, 2}:

-00*11*22* es una e.r. que denota el lenguaje cuyas cadenas están formadas por una cantidad $n \geq 1$ de 0s seguidas por una cantidad $m \geq 1$ de 1s y terminadas con una cantidad $ñ \geq 1$ de 2s.

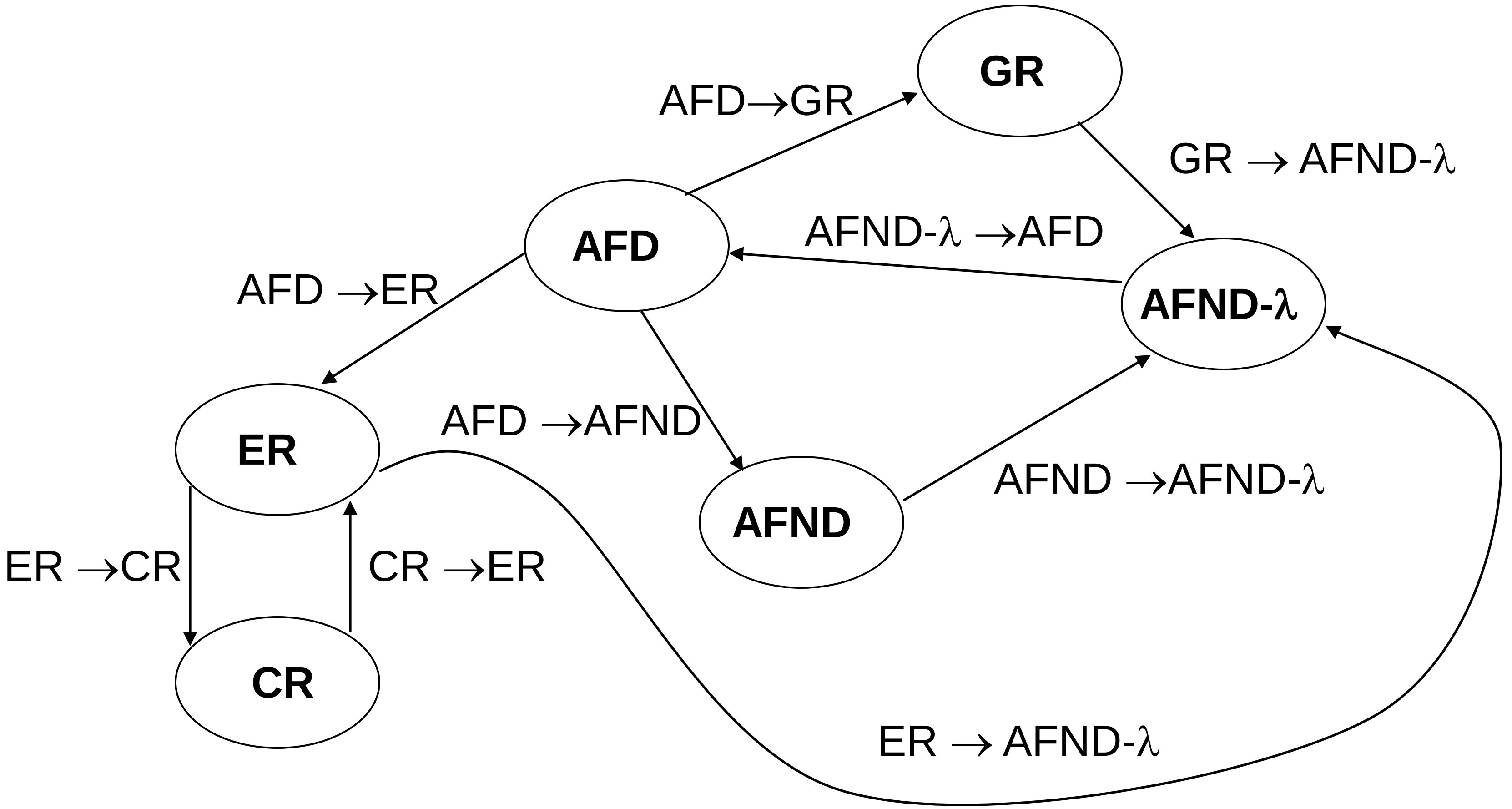
Definición:

Sea un AFD = < K, Σ, δ, q0, F > definimos **función de transición extendida** o $\hat{\delta}(q, \alpha)$ como

$$\hat{\delta}(q,\alpha) = \begin{cases} \delta(q,a) & \text{si} \quad \alpha = a \\ \hat{\delta}(\delta(q,a),\alpha') & \text{si} \quad \alpha = a\alpha' \end{cases}$$

donde α, α' $\in\Sigma^*$.

Equivalencias entre las representaciones de los lenguajes regulares



Pasaje de AFD a expresión regular

Teorema:

Si L es aceptado por un AFD M, entonces L puede ser expresado mediante una expresión regular.

Demostración (constructiva):

Sea L el lenguaje aceptado por $M = < \{ q_1, q_2,..., q_n \}, \Sigma, \delta, q_1, F >$.

Definimos el conjunto R_{ij}^k como aquel formado por todas las cadenas tal que

$\delta(q_i,x) = q_j$,

$\delta(q_i,y) = q_h$,

y es prefijo de x ($y \neq x$, $y \neq \lambda$)

$h \leq k$ con k:0..n.

Esto es, R_{ij}^{k} es el conjunto de cadenas que permiten ir del estado q_i al estado q_j y que si pasan por un estado diferente al q_i y al q_j, el subíndice de ese estado es menor o igual a k.

Nota: i y j pueden ser mayores que k. La restricción es sólo para los estados intermedios, en el caso que los haya.

Fijarse que, en realidad, los que nos interesa son los conjuntos R_{ij}^{n}

donde $q_j \in F$, puesto que el lenguaje aceptado por el autómata es

$$L(M) = \bigcup_{q_j \in F} R_{1j}^{n}$$

donde el estado q_1 es el estado inicial.

Los demás conjuntos serán útiles debido a que son un recurso para calcular los R_{ij}^{n}.

Como no hay estados con índice mayor que n, sólo interesan los conjuntos de cadenas R_{ij}^{k} con $k \leq n$.

Definamos formalmente R_{ij}^{k} :

$$R_{ij}^{k} = \begin{cases} R_{ij}^{0} = \begin{cases} \{a / \delta(q_i, a) = q_j\} & \text{si} \quad i \neq j \\ \{a / \delta(q_i, a) = q_j\} \cup \lambda & \text{si} \quad i = j \end{cases} \\ R_{ij}^{k} = R_{ik}^{k-1}\left(R_{kk}^{k-1}\right)^{*} R_{kj}^{k-1} \cup R_{ij}^{k-1} \quad \text{si} \quad k > 0, k \leq n \end{cases}$$

La parte 2 de la definición dice que las cadenas que pertenezcan a R_{ij}^{k} serán aquellas que ya pertenecían a R_{ij}^{k-1} unión las concatenaciones de:

- las cadenas que permitan ir del estado q_i al estado q_k sin pasar por un estado cuyo subíndice sea mayor a k-1,

- la clausura de las cadenas que permitan pasar del estado q_k al q_k sin pasar por un estado cuyo subíndice sea mayor a k-1, y

- las cadenas que permitan pasar del estado q_k al q_j sin pasar por un estado cuyo subíndice sea mayor a k-1.

De esta manera, sobre la información que se tiene de R^{k-1} construimos R_{ij}^{k}.

Para demostrar que las expresiones que corresponden a R_{ij}^k se pueden construir a partir de las expresiones obtenidas de R^{k-1} usamos inducción sobre k.

<u>Caso base</u> (k = 0)

R_{ij}^0 es un conjunto finito de cadenas las cuales son λ o un elemento del alfabeto y e_{ij}^0 la e.r. asociada a dicho conjunto.

Así, e_{ij}^0 será $a_1/a_2/../a_p$ (o $a_1/a_2/../a_p/\ \lambda$ si i = j)

donde $a_1..a_p$ son símbolos tal que $\delta(q_i, a_1) = q_j$ o $\delta(q_i, a_2) = q_j$ o ..

.. o $\delta(q_i, a_p) = q_j$. Si p = 0 $\Rightarrow$ ($e_{ij} = \phi \vee e_{ij} = \lambda$ si i = j).

<u>Paso inductivo</u> (asumimos que $\exists$ una expresión e_{lm}^{k-1} para cada R_{lm}^{k-1})

La expresión e_{ij}^k para R_{ij}^k estará dada por $(e_{ik}^{k-1})(e_{kk}^{k-1})^*(e_{kj}^{k-1})/e_{ij}^{k-1}$

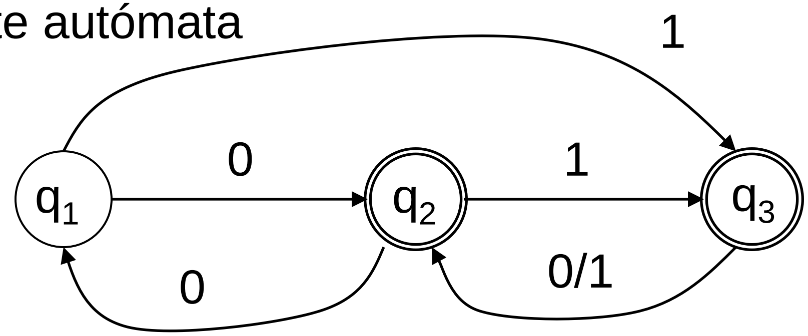
y aplicando unión, concatenación y clausura de los conjuntos de cadenas que forman los e^{k-1} obtenemos la expresión para e_{ij}^k.

Una vez obtenidos los e_{ij}^{k} $\forall$ k,i,j, el lenguaje aceptado por M estará dado por $L(M) = \bigcup_{q_j \in F} R_{1j}^{n}$

lo cual corresponderá a la expresión regular $e_{1j_1}^{n} / e_{1j_2}^{n} / .. / e_{1j_p}^{n}$ donde $\{q_{j_1}, q_{j_2}, .., q_{j_p}\} = F$.

Ejemplo:

Sea M el siguiente autómata



Encuentre una expresión regular que exprese el mismo lenguaje aceptado por M.

Veamos primero los valores de los conjuntos R^0 cuyas expresiones regulares asociadas son

$R^0_{11} = \{\lambda\}$ e.r. $\rightarrow \lambda$ $\quad$ $R^0_{21} = \{0\}$ e.r. $\rightarrow 0$ $\quad$ $R^0_{31} = \phi$ e.r. $\rightarrow \phi$

$R^0_{12} = \{0\}$ e.r. $\rightarrow 0$ $\quad$ $R^0_{22} = \{\lambda\}$ e.r. $\rightarrow \lambda$ $\quad$ $R^0_{32} = \{0,1\}$ e.r. $\rightarrow 0/1$

$R^0_{13} = \{1\}$ e.r. $\rightarrow 1$ $\quad$ $R^0_{23} = \{1\}$ e.r. $\rightarrow 1$ $\quad$ $R^0_{33} = \{\lambda\}$ e.r. $\rightarrow \lambda$

Calculemos, por ejemplo $R_{22}{}^{1}$

$$R_{22}^{1} = R_{21}^{0}\left(R_{11}^{0}\right)^{*} R_{12}^{0} \cup R_{22}^{0} = \{0\} \circ \{\lambda\}^{*} \circ \{0\} \cup \{\lambda\}$$

y por lo tanto la expresión regular correspondiente es $e_{22}^{1} = 00/\lambda$.

Otro ejemplo $R_{12}{}^{1}$:

$$R_{12}^{1} = R_{11}^{0}\left(R_{11}^{0}\right)^{*} R_{12}^{0} \cup R_{12}^{0} = \{\lambda\} \circ \{\lambda\}^{*} \circ \{0\} \cup \{0\}$$

con la expresión $e_{12}^{1} = 0$.

Otro ejemplo $R_{23}{}^{1}$:

$$R_{23}^{1} = R_{21}^{0}\left(R_{11}^{0}\right)^{*} R_{13}^{0} \cup R_{23}^{0} = \{0\} \circ \{\lambda\}^{*} \circ \{1\} \cup \{1\}$$

con la expresión $e_{23}^{1} = 01/1$ . .

Recordemos que $e_{12}^{1} = 0$

$$e_{22}^{1} = 00/\lambda$$

$$e_{23}^{1} = 01/1$$

Otro ejemplo R_{13}^{2}:

$$R_{13}^{2} = R_{12}^{1}\left(R_{22}^{1}\right)^{*} R_{23}^{1} \cup R_{13}^{1} = \{0\} \circ \{00, \lambda\}^{*} \circ \{01, 1\} \cup \{1\}$$

con la expresión

$$e_{13}^{2} = 0(00)^{*}(01/1)/1$$

Ver que (01/1) = (λ/0)1 y que (00)*(λ/0)=0*.

De este modo, 0(00)*(λ/0)1/1 = 00*1/1 = 0*1.

Veamos, lo que nos interesa a nosotros es R_{12}^3 y R_{13}^3 ya que L(M) $R_{12}^3 \cup R_{13}^3$.

Tenemos que

	$k = 1$	$k = 2$
e_{11}^k	λ	$(00)*$
e_{12}^k	0	$0(00)*$
e_{13}^k	1	$0*1$
e_{21}^k	0	$0(00)*$
e_{22}^k	$\lambda / 00$	$(00)*$
e_{23}^k	$1/01$	$0*1$
e_{31}^k	ϕ	$(0/1)(00)*0$
e_{32}^k	$0/1$	$(0/1)(00)*$
e_{33}^k	λ	$\lambda/(0/1)0*1$

Vamos a proceder operando directamente sobre las expresiones

$$e_{12}^{3} = e_{13}^{2}\left(e_{33}^{2}\right)^{*} e_{32}^{2} / e_{12}^{2} =$$

$$e_{12}^{3} = 0^{*}1(\lambda/(0/1)0^{*}1)^{*}(0/1)(00)^{*}/0(00)^{*} =$$

$$e_{12}^{3} = 0^{*}1((0/1)0^{*}1)^{*}(0/1)(00)^{*}/0(00)^{*}$$

Por otro lado,

$$e_{13}^{3} = 0^{*}1((0/1)0^{*}1)^{*}$$

Finalmente,

$$L(M) = e_{12}^{3} / e_{13}^{3} = 0^{*}1((0/1)0^{*}1)^{*}(\lambda/(0/1)(00)^{*})/0(00)^{*}$$

Pasaje de expresión regular a AFD

Dada un lenguaje **L** deseamos remover el primer símbolo **a** de las cadenas que forman parte de **L**. A esta operación la llamaremos «derivar **L** respecto de **a**» y la definimos según:

1. $\partial_{\lambda}(L) = L$
2. $\partial_{a}(L) = \{ \alpha \ / \ a\alpha \in L\}$ donde $a \in \Sigma$, $\alpha \in \Sigma^*$
3. $\partial_{w}(L) = \partial_{u}(\partial_{a}(L))$ si $w = a.u$, $a \in \Sigma$, $u \in \Sigma^*$

Ejemplos:

Sea $L = \{a^{3n} \ / \ n \in \mathbb{N}\} = \{aaa, aaaaaa, aaaaaaaaa, \ldots\}$

$\partial_{a}(L) = \{a^{3n-1} \ / \ n \in \mathbb{N}\} = \{aa, aaaaa, aaaaaaaa, \ldots\}$

$\partial_{b}(L) = \phi$

$\partial_{aaa}(L) = L \cup \{\lambda\}$

Para un lenguaje L definimos T(L) según:

$T(L) = \phi$ si L no contiene a λ

y

$T(L) = \{\lambda\}$ si L contiene a λ

Derivada de una expresión regular

De forma análoga podemos definir la derivada de una expresión regular según:

1. $\partial_\lambda(u) = u$, para u expresión regular
2. Para cada $a \in \Sigma$

   $\partial_a(\phi) = \phi$

   $\partial_a(\lambda) = \phi$

   $\partial_a(b) = \lambda$ si a = b,

   $\partial_a(b) = \phi$ si a ≠ b
3. Si u y v son dos expresiones regulares

   $\partial_a(u/v) = \partial_a(u) / \partial_a(v)$

   $\partial_a(u.v) = \partial_a(u).v$ / $T(u).\partial_a(v)$

   $\partial_a(u^*) = \partial_a(u).u^*$

donde T(u) está definida según:

$T(u) = \phi$ si $L(u)$ no contiene a λ

y

$T(u) = \{\lambda\}$ si $L(u)$ contiene a λ

4. Sea $a \in \Sigma$, $u \in \Sigma^*$

$$\partial_{ax}(u) = \partial_x(\partial_a(u))$$

Ejemplos:

$\partial_a(a) = \lambda$

$\partial_a(a \,/\, b) = \partial_a(a) \,/\, \partial_a(b) = \lambda \,/\, \phi = \lambda$

$\partial_a((a \,/\, b).c) = \partial_a(a \,/\, b).c \,/\, T(a \,/\, b).\ \partial_a(c) = \lambda.c \,/\, \phi.\ \phi = c$

$\partial_a(a^*) = a^*$

$\partial_a(a^+) = \partial_a(a.a^*) = \partial_a(a).a^* \,/\, T(a).\ \partial_a(a^*) = \lambda.a^* \,/\, \phi.a^* = a^*$

Consideremos el AFD $M = < K, \Sigma, \delta, q_0, F >$ que acepta el lenguaje

$L = L(M)$

y sea $\Sigma = \{a, b\}$.

Desde q_0, M aceptará el lenguaje L, pero desde q_0 existe, al menos, una transición por a, o por b, o dos transiciones, una para a y otra para b.
Fijarse que, luego de una transición por a, el autómata M evoluciona a un estado q_i, desde el cual aceptará el lenguaje $\partial_a(L)$.
Análogamente, luego de una transición por b, el autómata M evoluciona a un estado q_k , desde el cual aceptará el lenguaje $\partial_b(L)$.

De esta forma, dada una expresión regular *u*, calculando sistemáticamente las derivadas de *u* para cada símbolo de Σ, y luego, nuevamente calculando sistemáticamente las derivadas de las derivada de u para cada símbolo de Σ, y así sucesivamente, podremos ir descubriendo la estructura del autómata que acepte el leguaje expresado por *u*.

Veamos un ejemplo:

Consideremos la expresión regular

$$u = a^+(b.a^*/ \lambda) / ba^+$$

Llamaremos L_0 a L(*u*) y lo consideramos el lenguaje aceptado desde q_0.
Ahora, calculamos la derivada de L_0 respecto de a, esto es $\partial_a(L_0)$.

$$\begin{aligned}
\partial_a(L_0) &= \partial_a(a^+(b.a^*/ \lambda) / ba^+) \\
&= \partial_a(a^+(b.a^*/ \lambda)) / \partial_a(ba^+) \\
&= [\partial_a(a^+).(b.a^*/ \lambda) / T(a^+). \partial_a(b.a^*/ \lambda)] / [\partial_a(b). a^+ / T(b).\partial_a(a^+)] \\
&= [a^*.(ba^*/ \lambda) / \phi. (\partial_a(b.a^*) / \partial_a(\lambda))] / [\phi. a^+/ \phi .a^*] \\
&= a^*.(ba^*/ \lambda) / \phi = a^*.(ba^*/ \lambda) = L_1
\end{aligned}$$

Esto es, desde el estado q_0, M acepta L_0, pero al hacer una transición a un próximo estado consumiendo *a*, desde dicho estado aceptará L_1 .
Llamaremos a dicho estado q_1.

Veamos ahora qué ocurre si desde q_0 el autómata M transiciona consumiendo *b*.

$$
\begin{aligned}
\partial_b(L_0) &= \partial_b(a^+(b.a^* / \lambda) / ba^+) \\
&= \partial_b(a^+(b.a^* / \lambda)) / \partial_b(ba^+) \\
&= [\partial_b(a^+).(b.a^* / \lambda) / T(a^+).\ \partial_b(b.a^* / \lambda)] \ / \ [\partial_b(b).\ a^+ / T(b).\partial_b(a^+)] \\
&= [(\partial_b(a).a^* / T(a).\ \partial_b(a^*)).(ba^*/ \lambda) / \phi.(\partial_b(b.a^*) / \partial_b(\lambda))] / [\lambda.\ a^+/ \phi .a^*] \\
&= [(\phi / \phi).\ (ba^*/ \lambda) / \phi] / [a^+ / \phi]= a^+ = L_2
\end{aligned}
$$

Ahora, seguimos desde L_1 , es decir, desde q_1 :

$$
\begin{aligned}
\partial_a(L_1) &= \partial_a(a^*(b.a^* / \lambda)) = \partial_a(a^*)(b.a^* / \lambda) / T(a^*)\ \partial_a((b.a^* / \lambda) \\
&= a^*(b.a^* / \lambda) / \lambda\ \phi = a^*(b.a^* / \lambda) = L_1
\end{aligned}
$$

$$
\begin{aligned}
\partial_b(L_1) &= \partial_b(a^*(b.a^* / \lambda)) = \partial_b(a^*)(b.a^* / \lambda) / T(a^*)\ \partial_b(b.a^* / \lambda) \\
&= \phi / (\lambda\ \partial_b(b.a^*) / \phi)) = \phi / (\partial_b(b).a^* / T(b).\ \partial_b(a^*)) = \\
&= \lambda\ .a^* / \phi.\ \phi = a^* = L_3
\end{aligned}
$$

Procedemos igual para L_2

$$\partial_a(L_2) = \partial_a(a^+) = \partial_a(a.a^*) = \partial_a(a).a^* \;/\; T(a).\ \partial_a(a^*) = \lambda.a^* \;/\; \phi = a^* = L_3$$

$$\partial_b(L_2) = \partial_b(a^+) = \phi = L_t$$

donde L_t es el lenguaje aceptado por el estado trampa.

Para L_3 y L_t tenemos

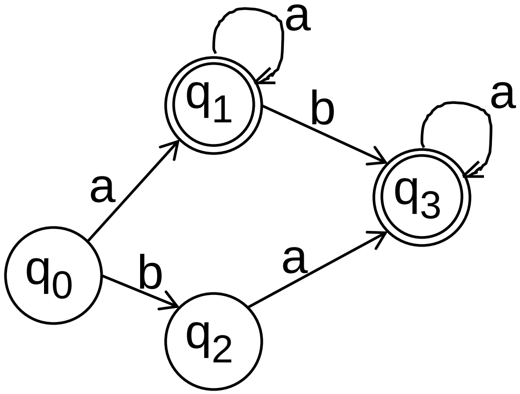
$$\partial_a(L_3) = \partial_a(a^*) = L_3$$

$$\partial_b(L_3) = \partial_b(a^*) = \phi$$

$$\partial_a(L_t) = \phi = L_t$$

$$\partial_b(L_t) = \phi = L_t$$

El autómata finito determinístico resultante es



donde los estados finales son aquellos que aceptan un lenguaje que contiene a λ.

Pasaje de expresión regular a AFND-λ

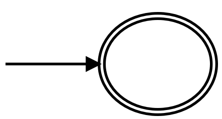
Teorema:

Sea r una e.r., entonces existe un AFND-λ que acepta L(r).

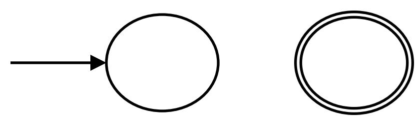
<u>Demostración</u> (por inducción sobre el número de operadores de r)

Caso base (sin operadores)

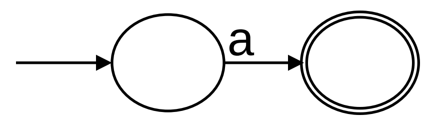
Sea r una expresión ϕ, λ o a / a$\in\Sigma$, el AFDN- λ correspondiente en cada caso es



r = λ



r = ϕ



r = a / a$\in\Sigma$

En cada caso el AFND-λ acepta exactamente a L(r).

<u>Paso inductivo</u> (asumimos que la HI es verdadera para expresiones regulares con menos de n operadores)

Caso 1:

Sea $r = r_1 / r_2$ tal que r_1 y r_2 tienen menos de n operadores. Luego, por HI existen $M_1 = < K_1, \Sigma_1, \delta_1, q_1, \{f_1\} >$ y $M_2 = < K_2, \Sigma_2, \delta_2, q_2, \{f_2\} >$ / M_1 y M_2 son AFND-λ y $L(M_1) = L(r_1)$ y $L(M_2) = L(r_2)$.

Se define M AFND-λ según

$M = < K_1 \cup K_2 \cup \{ q_0, f_0 \}, \Sigma_1 \cup \Sigma_2, \delta, q_0, \{ f_0 \} >$

donde δ está definido por

i) $\delta(q_0, \lambda) = \{q_1, q_2\}$,

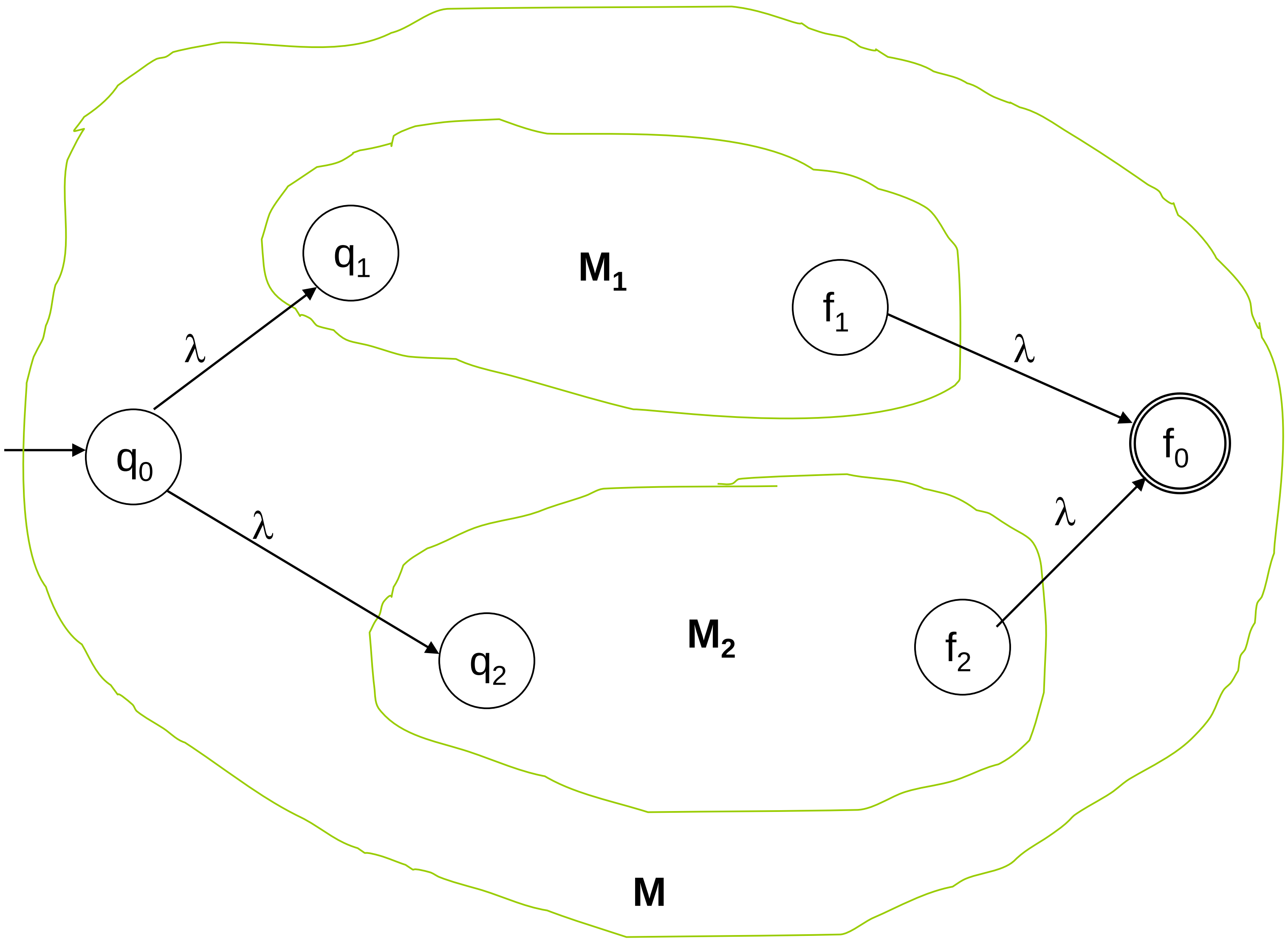
ii) $\delta(q, a) = \delta_1(q, a)$ si $q \in K_1 - \{ f_1 \} \wedge a \in \Sigma_1 \cup \{ \lambda \}$,

iii) $\delta(q, a) = \delta_2(q, a)$ si $q \in K_2 - \{ f_2 \} \wedge a \in \Sigma_2 \cup \{ \lambda \}$,

iv) $\delta(f_1, \lambda) = \delta(f_2, \lambda) = \{ f_0 \}$.

y los estados q_0 y f_0 son el estado inicial y final de M, respectivamente.

Luego L(M) = L(r). **Ejercicio: Demostrar**



$$\boxed{r = r_1 \,/\, r_2 \wedge L(r_1) = L(M_1) \wedge L(r_2) = L(M_2),\ L(r) = L(M)}$$

Caso 2:

Sea $r = r_1\, r_2$ tal que r_1 y r_2 tienen menos de n operadores. Luego, por HI existen $M_1 = < K_1, \Sigma_1, \delta_1, q_1, \{ f_1 \} >$ y $M_2 = < K_2, \Sigma_2, \delta_2, q_2, \{ f_2 \} >$ / M_1 y M_2 son AFND-λ y $L(M_1) = L(r_1)$ y $L(M_2) = L(r_2)$.

Se define M el AFND-λ según

$M = < K_1 \cup K_2, \Sigma_1 \cup \Sigma_2, \delta, q_1, \{ f_2 \} >$

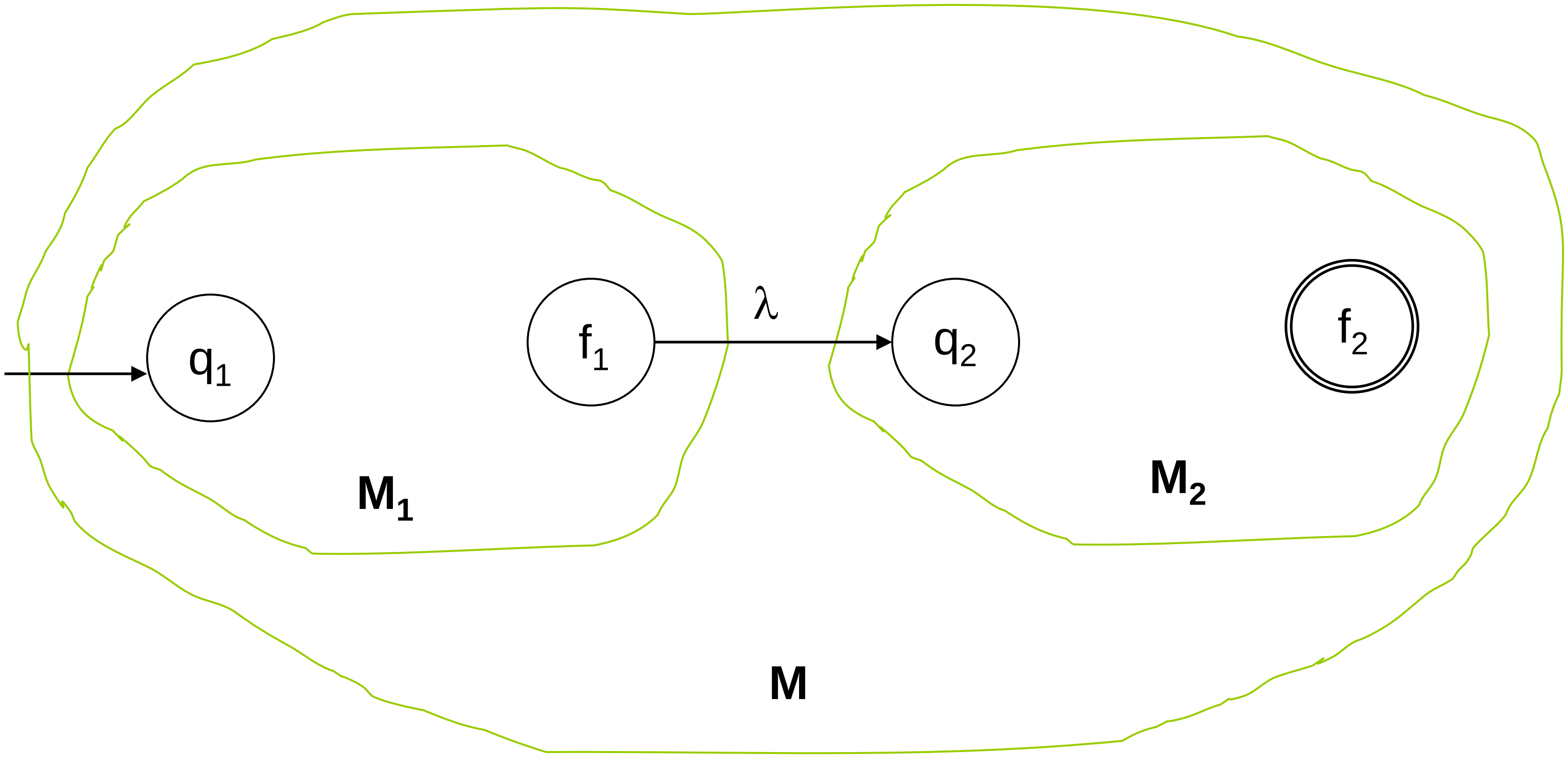
donde δ está definido por

i) $\delta(q, a) = \delta_1(q, a)$ si $q \in K_1 - \{ f_1 \} \wedge a \in \Sigma_1 \cup \{ \lambda \}$,

ii) $\delta(q, a) = \delta_2(q, a)$ si $q \in K_2 - \{ f_2 \} \wedge a \in \Sigma_2 \cup \{ \lambda \}$,

iii) $\delta(f_1, \lambda) = q_2$.

Luego $L(M) = L(r)$. **Ejercicio: Demostrar**



$r = r_1\ r_2$, $L(r_1) = L(M_1)$, $L(r_2) = L(M_2)$, $L(r) = L(M)$

Caso 3:

Sea $r = r_1^*$ tal que r_1 tienen menos de n operadores. Luego, por HI existe $M_1 = < K_1, \Sigma_1, \delta_1, q_1, \{f_1\} > / M_1$ AFND-λ y $L(M_1) = L(r_1)$.

Se define M el AFND-λ según

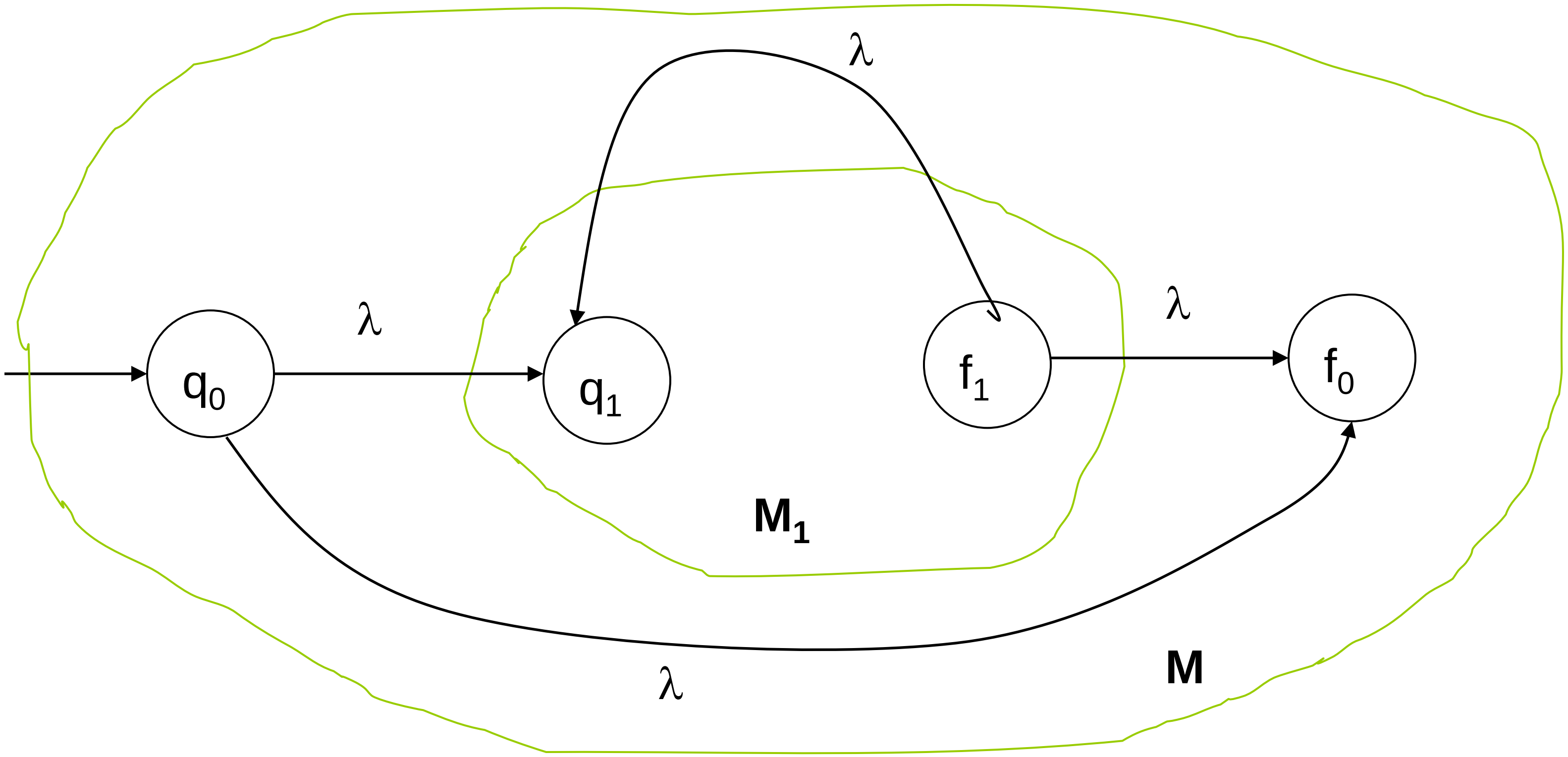
$M = < K1 \cup \{ q_0, f_0 \}, \Sigma_1, \delta, q_0, \{ f_0 \} >$

donde δ está definido por

i) $\delta(q, a) = \delta_1(q, a)$ si $q \in K_1 - \{ f_1 \} \wedge a \in \Sigma_1 \cup \{ \lambda \}$,

ii) $\delta(q_0, \lambda) = \delta(f_1, \lambda) = \{ q_1, f_0 \}$.

Luego $L(M) = L(r)$. **Ejercicio: Demostrar**



$r = r_1^* \wedge L(r_1) = L(M_1), L(r) = L(M)$

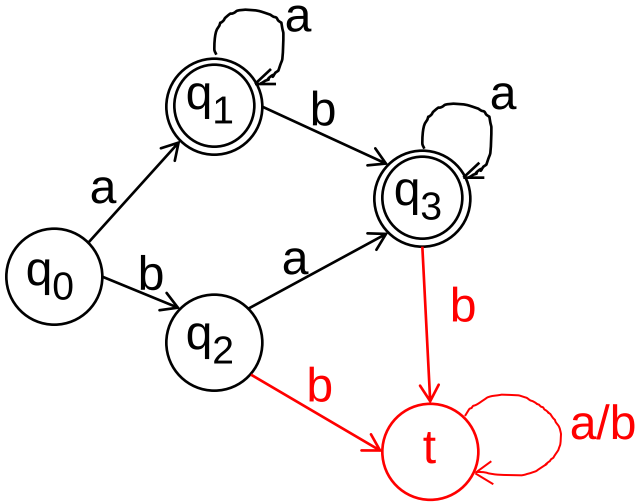
Pasaje de AFD a expresión regular

Tal como mencionamos anteriormente, dado un AFD M con estado inicial q_0, el lenguaje aceptado por M desde q_0, lo podemos denotar con una expresión regular.
Supongamos que en M, $\Sigma = \{ a, b \}$ y notemos L_0 como el lenguaje aceptado por M desde q_0.
Sea $\delta(q_0, a) = q_1$,
ya vimos que el lenguaje aceptado por M desde q_1 lo podemos denotar como $\partial_a(L_0)$ y
sea $\delta(q_0,b) = q_2$, el lenguaje aceptado por M desde q_2 lo podemos denotar como $\partial_b(L_0)$.
Así, podemos notar

$$L_0 = aL_1 / bL_2$$

donde L_1 es el lenguaje aceptado por M desde q_1 y L_2 desde q_2.

Sea el siguiente AFD M:



Podemos plantear las siguientes ecuaciones:

$L_0 = a.L_1 \,/\, b.L_2$

$L_1 = a.L_1 \,/\, b.L_3 \,/\, \lambda$

$L_2 = a.L_3 \,/\, b.L_t$

$L_3 = a.L_3 \,/\, b.L_t \,/\, \lambda$

$L_t = a.L_t \,/\, b.L_t \,/\, \phi$

¿Cómo resolvemos este sistema de ecuaciones?

Lema de Arden

Sean R, S y T expresiones regulares

Si R = S.R / T y $\lambda \notin$ S entonces

R = S*.T

Demostración (para toda cadena x)

i) L(S.R / T) $\subseteq$ L(S*.T)

<u>Caso base</u> (x = λ)

Si x $\in$ L(S.R / T) entonces x $\in$ T ya que $\lambda \notin$ S.

Pero si x $\in$ T entonces x $\in$ S*.T

<u>Paso inductivo</u> (la HI se cumple para todo x / |x| $\leq$ n)

Sea x / |x| = n + 1

Si x $\in$ L(S.R / T) entonces x $\in$ L(S.R) o x $\in$ L(T).

Si x $\in$ L(T) entonces x $\in$ L(S*.T).

Si x $\in$ L(S.R), entonces, sea x = yw / y $\in$ L(S) y w $\in$ L(R).

Como $\lambda \notin$ S entonces |y| > 0 y por lo tanto |x| > |w|.

Entonces, si w pertenece a L(R), por HI w $\in$ L(S*.T) y

x $\in$ L(SS*.T) y como L(SS*.T) $\subseteq$ L(S*T), entonces

x $\in$ L(S*.T).

ii) $L(S^*.T) \subseteq L(S.R / T)$ o $L(S^i.T) \subseteq L(S.R / T)$, $i \geq 0$.

Caso base ($i = 0$)

Si $x \in L(S^0.T) \Rightarrow x \in L(T) \Rightarrow x \in L(S.R / T)$

Paso inductivo (la HI es verdadera para $i < n$)

Si $x \in L(S^n.T) \Rightarrow x \in L(S).L(S^{n-1}.T) \Rightarrow x \in L(S).L(R) \Rightarrow$
$x \in L(S.R) \Rightarrow x \in L(S.R / T)$

Volviendo al sistema de ecuaciones del ejemplo:

$L_0 = a.L_1 / b.L_2$

$L_1 = a.L_1 / b.L_3 / \lambda$

$L_2 = a.L_3 / b.L_t$

$L_3 = a.L_3 / b.L_t / \lambda$

$L_t = a.L_t / b.L_t / \phi$

podemos resolverlo usando la propiedad enunciada en el Lema de Arden haciendo:

$L_t = (a/b). L_t / \phi$, y como $\lambda \notin (a/b)$ entonces $L_t = (a/b)^*. \phi = \phi$

$L_3 = a.L_3 / b.L_t / \lambda = a.L_3 / \lambda = a^*. \lambda = a^*$

$L_2 = a.L_3 / b.L_t = a.L_3 = a.a^* = a^+$

$L_1 = a.L_1 / b.L_3 / \lambda = a^*.(b. L_3 / \lambda) = a^*.(b.a^* / \lambda)$

$L_0 = a.L_1 / b.L_2 = a. a^*.(b.a^* / \lambda) / b. a^+ = a^+.(b.a^* / \lambda) / b. a^+$

Pasaje de AFD a GR

Algortimo

Entrada: AFD $M = < K, \Sigma, \delta, q_0, F >$

Salida : $G = < V_N, V_T, P, Q_0 >$

1. Hacer $V_T \leftarrow \Sigma$
2. Hacer $V_N \leftarrow K$ (con sus estados escritos en mayúsculas)
3. Para cada estado $q \in K$ y para cada $a \in \Sigma$

   Si $r = \delta(q, a)$

   agregar a P, $(Q \rightarrow a\ R)$, $Q,R \in V_N$

   si $r \in F$ agregar a P, $(Q \rightarrow a)$

4. Si $q_0 \in F$

   agregar a P $(Q_0 \rightarrow \lambda)$

Lema:

Sea el AFD M = < K, Σ, δ, q_0, F > y G = < V_N, Σ, P, S > una gramática

regular, luego $(q, \alpha) \vdash^* (r, \lambda) \Leftrightarrow Q \rightarrow^* \alpha R$ donde los símbolos no terminales se notan como los estados de K pero en mayúsculas.

Demostración (por inducción en la longitud de α):

Caso base ($\alpha = \lambda$)

$(q, \lambda) \vdash^* (r, \lambda) \Leftrightarrow (q, \lambda) = (r, \lambda) \Leftrightarrow q = r \Leftrightarrow Q \rightarrow \lambda R$

Nota: siempre es cierto $Q \rightarrow \lambda R$ con R = Q pues basta no aplicar

ninguna producción, así como en el AFD $(q, \lambda) \vdash (r, \lambda) \Rightarrow q = r$.

Paso inductivo ($\alpha = a\ \alpha'$)

$$(q, a\alpha') \vdash^* (r, \lambda) \Leftrightarrow (q, a\alpha') \vdash (q', \alpha') \vdash^* (r, \lambda) \Leftrightarrow$$

def HI

$$\Leftrightarrow Q \rightarrow aQ' \wedge Q' \rightarrow \alpha' R \Leftrightarrow Q \rightarrow aQ' \rightarrow a\alpha' R \Leftrightarrow Q \rightarrow \alpha R$$

Teorema:

Sean M y G como los dados en el algoritmo precedente,

entonces L(M) = L(G).

<u>Demostración</u>:

i) L(M) $\subseteq$ L(G)

Si $\alpha \in L(M) \Rightarrow (q_0, \alpha) \models^{+} (f, \lambda)$, $f \in F$.

Si $\alpha = \lambda \Rightarrow q_0 \in F$, pero en ese caso existe una producción $Q_0 \rightarrow \lambda$ (paso 4 del algoritmo).

Si $\alpha \neq \lambda \Rightarrow \alpha = \alpha' a$ y podemos escribir $(q_0, \alpha) \models^{+} (f, \lambda)$ como $(q_0, \alpha' a) \models^{*} (q', a) \models (f, \lambda)$.

Por el lema anterior, existe $Q_0 \xrightarrow{*} \alpha' Q'$.

Como $(q', a) \models (f, \lambda) \Rightarrow f = \delta(q', a)$, $f \in F$

entonces existe una derivación $Q' \rightarrow a$.

Así que, $Q_0 \rightarrow \alpha' Q' \rightarrow \alpha' a \Rightarrow Q_0 \rightarrow \alpha \Rightarrow \alpha \in L(G)$.

ii) $L(G) \subseteq L(M)$

Si $\alpha \in L(G) \Rightarrow Q_0 \rightarrow^* \alpha$.

Si $\alpha = \lambda \Rightarrow$ existe en G, $Q_0 \rightarrow \lambda$, luego $q_0 \in F \Rightarrow \lambda \in L(M)$.

Si $\alpha \neq \lambda \Rightarrow \alpha = \alpha'a$, entonces $Q_0 \rightarrow \alpha'Q' \rightarrow \alpha'a$, dado que $Q' \rightarrow a$.

Por el lema anterior, $Q_0 \rightarrow \alpha'Q' \Rightarrow (q_0, \alpha'a) \models^* (q', a)$.

Pero $Q' \rightarrow a \Rightarrow f = \delta(q', a)$, $f \in F$, luego existe en M $(q', a) \models (f, \lambda)$.

Así $(q_0, \alpha) = (q_0, \alpha'a) \models^* (q', a) \models (f, \lambda)$ que es lo mismo que

$(q_0, \alpha) \models^+ (f, \lambda) \Rightarrow \alpha \in L(M)$.

Pasaje de GR a AFND-λ

Algoritmo

Entrada: $G = \langle V_N, V_T, P, S \rangle$

Salida: $M = \langle K, \Sigma, \delta, q_0, F \rangle$

1. Hacer $K \leftarrow V_N \cup \{ f \}$
2. Para cada derivación $A \rightarrow aB \in P$

   Agregar B a $\delta(A, a)$

3. Para cada derivación $A \rightarrow a$

   Agregar f a $\delta(A, a)$

4. Si existe derivación $S \rightarrow \lambda$

   Agregar f a $\delta(S, \lambda)$