Equivalencia entre Lenguajes Regulares y Autómatas Finitos (Teorema de Kleene)

Introducción a la Lógica y la Computación (3era Parte)

Docentes: Badano, Bustos, Costamagna, Tellechea, Zigaran

Año 2024

Equivalencia entre Lenguajes Regulares y Autómatas Finitos

Un lenguaje es regular si y solo si es aceptado por un AF.

Equivalencia entre Lenguajes Regulares y Autómatas Finitos

Un lenguaje es regular si y solo si es aceptado por un AF.

"Una interpretación posible de este resultado es que los AF son las máquinas secuenciales que caracterizan a los lenguajes regulares."

Equivalencia entre Lenguajes Regulares y Autómatas Finitos

Un lenguaje es regular si y solo si es aceptado por un AF.

"Una interpretación posible de este resultado es que los AF son las máquinas secuenciales que caracterizan a los lenguajes regulares."

- Ya probamos que $LR^{\Sigma} \equiv ER^{\Sigma}$ y $AFD^{\Sigma} \equiv AFN^{\Sigma} \equiv AFN\epsilon^{\Sigma}$.

Equivalencia entre Lenguajes Regulares y Autómatas Finitos

Un lenguaje es regular si y solo si es aceptado por un AF.

"Una interpretación posible de este resultado es que los AF son las máquinas secuenciales que caracterizan a los lenguajes regulares."

- Ya probamos que $LR^{\Sigma} \equiv ER^{\Sigma}$ y $AFD^{\Sigma} \equiv AFN^{\Sigma} \equiv AFN\epsilon^{\Sigma}$.
- Uniremos esta dos cadenas de equivalencia probando que $ER^{\Sigma} \equiv AF^{\Sigma}$.

Equivalencia entre Lenguajes Regulares y Autómatas Finitos

Un lenguaje es regular si y solo si es aceptado por un AF.

"Una interpretación posible de este resultado es que los AF son las máquinas secuenciales que caracterizan a los lenguajes regulares."

- Ya probamos que $LR^{\Sigma} \equiv ER^{\Sigma}$ y $AFD^{\Sigma} \equiv AFN^{\Sigma} \equiv AFN\epsilon^{\Sigma}$.
- Uniremos esta dos cadenas de equivalencia probando que $ER^{\Sigma} \equiv AF^{\Sigma}$.

Teorema de Kleene ($ER^{\Sigma} \equiv AF^{\Sigma}$)

1. *(ida) Si $e \in ER^{\Sigma}$, entonces $\exists M \in AFN\epsilon^{\Sigma}$ tq $L(M) = L(e)$.*
2. *(vuelta) Si $M \in AFN^{\Sigma}$, entonces $\exists e \in ER^{\Sigma}$ tq $L(M) = L(e)$.*

Para probar el inciso 1, notar que:

Para probar el inciso 1, notar que:

- Un AF puede ser visto como una estructura recursiva: es un estado conectado con n sub-autómatas.

Para probar el inciso 1, notar que:

- Un AF puede ser visto como una estructura recursiva: es un estado conectado con n sub-autómatas.
- Por lo tanto, podemos construir un AF recursivamente acompañando la recursividad de la expresiones regulares.

Demo 1.

Hacemos inducción en la forma de la expresión regular e.

Demo 1.

Hacemos inducción en la forma de la expresión regular e.

Casos Bases:

Si $e = \emptyset$, entonces tomamos M:

y luego, $L(M) = \emptyset = L(\emptyset)$.

Demo 1.

Hacemos inducción en la forma de la expresión regular e.

Casos Bases:

Si $e = \emptyset$, entonces tomamos M:

y luego, $L(M) = \emptyset = L(\emptyset)$.

Si $e = \epsilon$, entonces tomamos M:

y luego, $L(M) = \{\epsilon\} = L(\epsilon)$.

Demo 1.
Hacemos inducción en la forma de la expresión regular e.

Casos Bases:

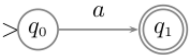
Si $e = \emptyset$, entonces tomamos M:

y luego, $L(M) = \emptyset = L(\emptyset)$.

Si $e = \epsilon$, entonces tomamos M:

y luego, $L(M) = \{\epsilon\} = L(\epsilon)$.

Si $e = a$, entonces tomamos M:



y luego, $L(M) = \{a\} = L(a)$.

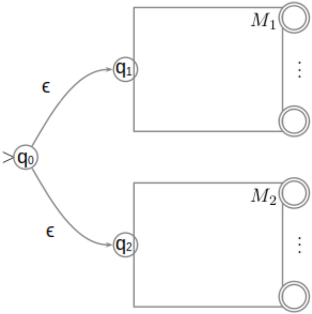
Casos Recursivos:

Si $e = e_1 + e_2$, por HI, existen $M_1 = (\Sigma, Q_1, q_1, F_1, \Delta_1)$ y $M_2 = (\Sigma, Q_2, q_2, F_2, \Delta_2)$ $AFN\epsilon$ tq $L(M_1) = L(e_1)$ y $L(M_2) = L(e_2)$ $\therefore$ podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ AFNϵ donde:

Casos Recursivos:

Si $e = e_1 + e_2$, por HI, existen $M_1 = (\Sigma, Q_1, q_1, F_1, \Delta_1)$ y $M_2 = (\Sigma, Q_2, q_2, F_2, \Delta_2)$ $AFN\epsilon$ tq $L(M_1) = L(e_1)$ y $L(M_2) = L(e_2)$ $\therefore$ podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ AFNϵ donde:

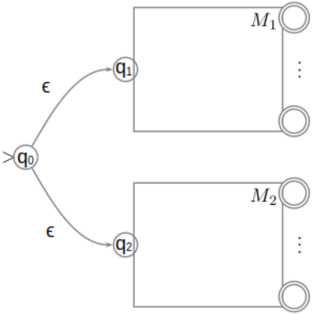
$Q = Q_1 \cup Q_2 \cup \{q_0\}$ con $q_0 \notin Q_1, Q_2$
$F = F_1 \cup F_2$
$\Delta = \Delta_1 \cup \Delta_2 \cup \{((q_0, \epsilon), \{q_1, q_2\})\}$



Casos Recursivos:

Si $e = e_1 + e_2$, por HI, existen $M_1 = (\Sigma, Q_1, q_1, F_1, \Delta_1)$ y $M_2 = (\Sigma, Q_2, q_2, F_2, \Delta_2)$ $AFN\epsilon$ tq $L(M_1) = L(e_1)$ y $L(M_2) = L(e_2)$ $\therefore$ podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ AFNϵ donde:

$Q = Q_1 \cup Q_2 \cup \{q_0\}$ con $q_0 \notin Q_1, Q_2$
$F = F_1 \cup F_2$
$\Delta = \Delta_1 \cup \Delta_2 \cup \{((q_0, \epsilon), \{q_1, q_2\})\}$

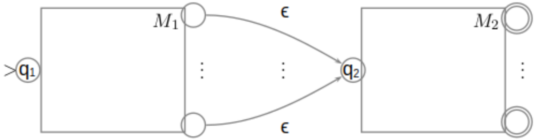


Luego, $L(M) = L(M_1) \cup L(M_2) = L(e_1) \cup L(e_2) = L(e_1 + e_2)$.

Si $e = e_1 e_2$, por HI, existen $M_1 = (\Sigma, Q_1, q_1, F_1, \Delta_1)$ y $M_2 = (\Sigma, Q_2, q_2, F_2, \Delta_2)$ $AFN\epsilon$ tq $L(M_1) = L(e_1)$ y $L(M_2) = L(e_2)$ $\therefore$ podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ AFNϵ donde:

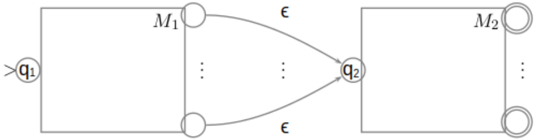
Si $e = e_1 e_2$, por HI, existen $M_1 = (\Sigma, Q_1, q_1, F_1, \Delta_1)$ y $M_2 = (\Sigma, Q_2, q_2, F_2, \Delta_2)$ $AFN\epsilon$ tq $L(M_1) = L(e_1)$ y $L(M_2) = L(e_2)$ $\therefore$ podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ AFNϵ donde:

$$
\begin{aligned}
&Q = Q_1 \cup Q_2 \\
&q_0 = q_1 \\
&F = F_2 \\
&\Delta = (\Delta_1 \cup \Delta_2) - \{((q, \epsilon), \Delta_1(q, \epsilon)) : q \in F_1\} \\
&\qquad\qquad \cup \{((q, \epsilon), \Delta_1(q, \epsilon) \cup \{q_2\}) : q \in F_1\}
\end{aligned}
$$



Si $e = e_1 e_2$, por HI, existen $M_1 = (\Sigma, Q_1, q_1, F_1, \Delta_1)$ y $M_2 = (\Sigma, Q_2, q_2, F_2, \Delta_2)$ $AFN\epsilon$ tq $L(M_1) = L(e_1)$ y $L(M_2) = L(e_2)$ $\therefore$ podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ AFNϵ donde:

$$
\begin{aligned}
Q &= Q_1 \cup Q_2 \\
q_0 &= q_1 \\
F &= F_2 \\
\Delta &= (\Delta_1 \cup \Delta_2) - \{((q, \epsilon), \Delta_1(q, \epsilon)) : q \in F_1\} \\
&\qquad \cup \{((q, \epsilon), \Delta_1(q, \epsilon) \cup \{q_2\}) : q \in F_1\}
\end{aligned}
$$

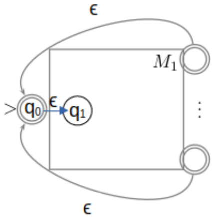


Luego, $L(M) = L(M_1)L(M_2) = L(e_1)L(e_2) = L(e_1 e_2)$.

Si $e = e_1{}^*$, por HI, existe $M_1 = (\Sigma, Q_1, q_1, F_1, \Delta_1)$ $AFN\epsilon$ tq $L(M_1) = L(e_1)$ $\therefore$ podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ AFNϵ donde:

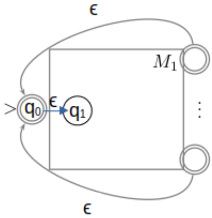
Si $e = e_1{}^*$, por HI, existe $M_1 = (\Sigma, Q_1, q_1, F_1, \Delta_1)$ $AFN\epsilon$ tq $L(M_1) = L(e_1)$ $\therefore$ podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ AFNϵ donde:

$$
\begin{aligned}
Q &= Q_1 \cup \{q_0\}, \text{ con } q_0 \notin Q_1 \\
F &= F_1 \cup \{q_0\} \\
\Delta &= \Delta_1 - \{((q,\epsilon), \Delta_1(q,\epsilon)) : q \in F_1\} \\
&\qquad \cup \{((q,\epsilon), \Delta_1(q,\epsilon) \cup \{q_0\}) : q \in F_1\} \\
&\qquad \cup \{((q0,\epsilon), \{q_1\})\}
\end{aligned}
$$



Si $e = e_1{}^*$, por HI, existe $M_1 = (\Sigma, Q_1, q_1, F_1, \Delta_1)$ $AFN\epsilon$ tq $L(M_1) = L(e_1)$ $\therefore$ podemos tomar $M = (\Sigma, Q, q_0, F, \Delta)$ AFNϵ donde:

$$
\begin{aligned}
&Q = Q_1 \cup \{q_0\}, \text{ con } q_0 \notin Q_1 \\
&F = F_1 \cup \{q_0\} \\
&\Delta = \Delta_1 - \{((q, \epsilon), \Delta_1(q, \epsilon)) : q \in F_1\} \\
&\qquad\quad \cup \{((q, \epsilon), \Delta_1(q, \epsilon) \cup \{q_0\}) : q \in F_1\} \\
&\qquad\quad \cup \{((q0, \epsilon), \{q_1\})\}
\end{aligned}
$$



Luego, $L(M) = (L(M_1))^* = (L(e_1))^* = L(e_1{}^*)$.

Algoritmo de Kleene (ida)

La demo anterior nos brinda un algoritmo recursivo que dada una expresión regular, devuelve un AF equivalente.

Algoritmo de Kleene (ida)

La demo anterior nos brinda un algoritmo recursivo que dada una expresión regular, devuelve un AF equivalente.

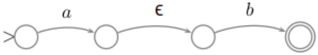
Por ejemplo, para la expresión regular $ab + ba$, hacemos:

Algoritmo de Kleene (ida)

La demo anterior nos brinda un algoritmo recursivo que dada una expresión regular, devuelve un AF equivalente.

Por ejemplo, para la expresión regular $ab + ba$, hacemos:

1) AF que acepta ab:

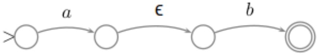


Algoritmo de Kleene (ida)

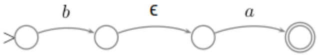
La demo anterior nos brinda un algoritmo recursivo que dada una expresión regular, devuelve un AF equivalente.

Por ejemplo, para la expresión regular $ab + ba$, hacemos:

1) AF que acepta ab:



2) AF que acepta ba:

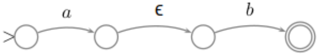


Algoritmo de Kleene (ida)

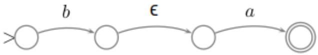
La demo anterior nos brinda un algoritmo recursivo que dada una expresión regular, devuelve un AF equivalente.

Por ejemplo, para la expresión regular $ab + ba$, hacemos:

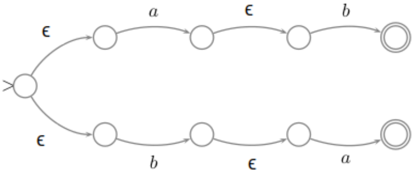
1) AF que acepta ab:



2) AF que acepta ba:



3) AF que acepta $ab + ba$:



Teorema de Kleene (vuelta)

Para probar la vuelta del Teorema de Kleene (2) necesitamos enunciar el siguiente lema:

Teorema de Kleene (vuelta)

Para probar la vuelta del Teorema de Kleene (2) necesitamos enunciar el siguiente lema:

Lema (de Arden)

Si A, B son lenguajes y $\epsilon \notin A$, entonces la ecuación $X = AX \cup B$ tiene solución única y es $X = A^*B$.

Teorema de Kleene (vuelta)

Para probar la vuelta del Teorema de Kleene (2) necesitamos enunciar el siguiente lema:

Lema (de Arden)

Si A, B son lenguajes y $\epsilon \notin A$, entonces la ecuación $X = AX \cup B$ tiene solución única y es $X = A^*B$.

Notar que si A y B son regulares, entonces:
1) X es regular
2) A, B y X pueden ser denotados por expresiones regulares

Teorema de Kleene (vuelta)

Para probar la vuelta del Teorema de Kleene (2) necesitamos enunciar el siguiente lema:

Lema (de Arden)

Si A, B son lenguajes y $\epsilon \notin A$, entonces la ecuación $X = AX \cup B$ tiene solución única y es $X = A^*B$.

Notar que si A y B son regulares, entonces:
1) X es regular
2) A, B y X pueden ser denotados por expresiones regulares

Entonces:

$X = aX + b^*ab$ tiene solución única $X = a^*b^*ab$.

Teorema de Kleene (vuelta)

Para probar la vuelta del Teorema de Kleene (2) necesitamos enunciar el siguiente lema:

Lema (de Arden)

Si A, B son lenguajes y $\epsilon \notin A$, entonces la ecuación $X = AX \cup B$ tiene solución única y es $X = A^*B$.

Notar que si A y B son regulares, entonces:
1) X es regular
2) A, B y X pueden ser denotados por expresiones regulares

Entonces:

$X = aX + b^*ab$ tiene solución única $X = a^*b^*ab$.

$X = a^2X + b^+X + ab = (a^2 + b^+)X + ab$ tiene solución única $X = (a^2 + b^+)^*ab$.

Teorema de Kleene (vuelta)

Para probar la vuelta del Teorema de Kleene (2) necesitamos enunciar el siguiente lema:

Lema (de Arden)

Si A, B son lenguajes y $\epsilon \notin A$, entonces la ecuación $X = AX \cup B$ tiene solución única y es $X = A^*B$.

Notar que si A y B son regulares, entonces:
1) X es regular
2) A, B y X pueden ser denotados por expresiones regulares

Entonces:

$X = aX + b^*ab$ tiene solución única $X = a^*b^*ab$.

$X = a^2X + b^+X + ab = (a^2 + b^+)X + ab$ tiene solución única $X = (a^2 + b^+)^*ab$.

$X = ab^2X + aX + a^*b + b^*a = (ab^2 + a)X + (a^*b + b^*a)$ tiene solución única $X = (ab^2 + a)^*(a^*b + b^*a)$.

Teorema de Kleene (vuelta)

Demo 2. Sea $M = (\Sigma, Q, q_0, F, \delta)$ AFN con $Q = \{q_0, q_1, \ldots, q_n\}$, definimos $M_i = (\Sigma, Q, q_i, F, \Delta)$ el AFN que se obtiene a partir de M cambiando únicamente el estado inicial de M por q_i.

Teorema de Kleene (vuelta)

Demo 2. Sea $M = (\Sigma, Q, q_0, F, \delta)$ AFN con $Q = \{q_0, q_1, \dots, q_n\}$, definimos $M_i = (\Sigma, Q, q_i, F, \Delta)$ el AFN que se obtiene a partir de M cambiando únicamente el estado inicial de M por q_i.

Y definimos $L(M_i) = X_i$, por lo tanto, $L(M) = X_0$.

Teorema de Kleene (vuelta)

Demo 2. Sea $M = (\Sigma, Q, q_0, F, \delta)$ AFN con $Q = \{q_0, q_1, \ldots, q_n\}$, definimos $M_i = (\Sigma, Q, q_i, F, \Delta)$ el AFN que se obtiene a partir de M cambiando únicamente el estado inicial de M por q_i.

Y definimos $L(M_i) = X_i$, por lo tanto, $L(M) = X_0$.

Cada X_i se puede expresar con la siguiente ecuación:

$$X_i = \begin{cases} \displaystyle\sum_{q_j \in \Delta(q_i, a)} aX_j & \text{si } q_i \notin F \\ \displaystyle\sum_{q_j \in \Delta(q_i, a)} aX_j + \epsilon & \text{si } q_i \in F \end{cases}$$

Por lo tanto, tenemos un sistema de $(n+1)$ ecuaciones por $(n+1)$ incógnitas al cual le aplicaremos el leman de Arden sucesivamente de la siguiente manera:

Por lo tanto, tenemos un sistema de $(n+1)$ ecuaciones por $(n+1)$ incógnitas al cual le aplicaremos el leman de Arden sucesivamente de la siguiente manera:

- Aplicamos Arden a la ecuación de X_n para obtener su valor y reemplazarlo en el resto de las ecuaciones reduciéndolo a un sistema $n \times n$.

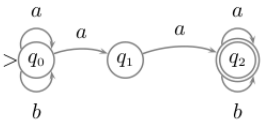
Por lo tanto, tenemos un sistema de $(n+1)$ ecuaciones por $(n+1)$ incógnitas al cual le aplicaremos el leman de Arden sucesivamente de la siguiente manera:

- Aplicamos Arden a la ecuación de X_n para obtener su valor y reemplazarlo en el resto de las ecuaciones reduciéndolo a un sistema $n \times n$.
- Repetimos lo anterior para la ecuación de X_{n-1}, y asi sucesivamente, hasta obtener un sistema de 1x1 con la ecuación correspondiente a X_0.

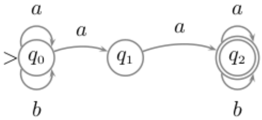
Por lo tanto, tenemos un sistema de $(n+1)$ ecuaciones por $(n+1)$ incógnitas al cual le aplicaremos el leman de Arden sucesivamente de la siguiente manera:

- Aplicamos Arden a la ecuación de X_n para obtener su valor y reemplazarlo en el resto de las ecuaciones reduciéndolo a un sistema $n \times n$.
- Repetimos lo anterior para la ecuación de X_{n-1}, y asi sucesivamente, hasta obtener un sistema de 1x1 con la ecuación correspondiente a X_0.
- Aplicamos por última vez Arden, obteniendo la solución final que buscabamos, pues $X_0 = L(M)$.

Por ejemplo, consideremos el siguiente AFN M:



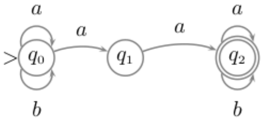
Por ejemplo, consideremos el siguiente AFN M:



El sistema de ecuaciones que determina el automata M es:

$$\begin{cases} X_0 = aX_0 + bX_0 + aX_1 & (1) \\ X_1 = aX_2 & (2) \\ X_2 = aX_2 + bX_2 + \epsilon & (3) \end{cases}$$

Por ejemplo, consideremos el siguiente AFN M:



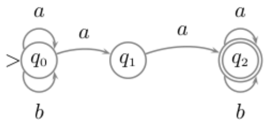
El sistema de ecuaciones que determina el automata M es:

$$\begin{cases} X_0 = aX_0 + bX_0 + aX_1 & (1) \\ X_1 = aX_2 & (2) \\ X_2 = aX_2 + bX_2 + \epsilon & (3) \end{cases}$$

Aplicamos Arden en la ecuacion (3):

$X_2 = aX_2 + bX_2 + \epsilon = (a + b)X_2 + \epsilon \therefore$
$X_2 = (a + b)^*$

Por ejemplo, consideremos el siguiente AFN M:



El sistema de ecuaciones que determina el automata M es:

$$\begin{cases} X_0 = aX_0 + bX_0 + aX_1 & (1) \\ X_1 = aX_2 & (2) \\ X_2 = aX_2 + bX_2 + \epsilon & (3) \end{cases}$$

Aplicamos Arden en la ecuacion (3):

$X_2 = aX_2 + bX_2 + \epsilon = (a+b)X_2 + \epsilon \therefore$
$X_2 = (a+b)^*$

Reemplazamos el valor obtenido en todo el sistema:

$$\begin{cases} X_0 = aX_0 + bX_0 + aX_1 & (1) \\ X_1 = a(a+b)^* & (2) \end{cases}$$

No es necesario aplicar Arden en la ecuacion (2) pues:

$X_1 = a(a + b)^*$

No es necesario aplicar Arden en la ecuacion (2) pues:

$X_1 = a(a+b)^*$

Reemplazamos el valor de X_1 obtenido en el resto del sistema:

$$\left\{ X_0 = aX_0 + bX_0 + a^2(a+b)^* \quad (1) \right.$$

No es necesario aplicar Arden en la ecuacion (2) pues:

$X_1 = a(a+b)^*$

Reemplazamos el valor de X_1 obtenido en el resto del sistema:

$$\begin{cases} X_0 = aX_0 + bX_0 + a^2(a+b)^* & (1) \end{cases}$$

Aplicamos Arden por última vez a la ecuación (1):

$X_0 = aX_0 + bX_0 + a^2(a+b)^* = (a+b)X_0 + a^2(a+b)^* \therefore$
$X_0 = (a+b)^* a^2 (a+b)^*$

No es necesario aplicar Arden en la ecuacion (2) pues:

$X_1 = a(a+b)^*$

Reemplazamos el valor de X_1 obtenido en el resto del sistema:

$$\{X_0 = aX_0 + bX_0 + a^2(a+b)^* \quad (1)$$

Aplicamos Arden por última vez a la ecuación (1):

$X_0 = aX_0 + bX_0 + a^2(a+b)^* = (a+b)X_0 + a^2(a+b)^* \therefore$
$X_0 = (a+b)^* a^2(a+b)^*$

Y esta última solución de X_0 es la expresión regular que denota al lenguaje aceptado por el autómata M, por lo tanto:

$L(M) = (a+b)^* a^2(a+b)^*$

Resumen

- Probamos que $ER^{\Sigma} \equiv AF^{\Sigma}$ (Teo. de Kleene)

Resumen

- Probamos que $ER^{\Sigma} \equiv AF^{\Sigma}$ (Teo. de Kleene)
- Por transitividad, tenemos toda la cadena de equivalencias probada:

$$LR^{\Sigma} \equiv ER^{\Sigma} \equiv AFD^{\Sigma} \equiv AFN^{\Sigma} \equiv AFN\epsilon^{\Sigma}$$

Resumen

- Probamos que $ER^{\Sigma} \equiv AF^{\Sigma}$ (Teo. de Kleene)
- Por transitividad, tenemos toda la cadena de equivalencias probada:

$$LR^{\Sigma} \equiv ER^{\Sigma} \equiv AFD^{\Sigma} \equiv AFN^{\Sigma} \equiv AFN\epsilon^{\Sigma}$$

- Por lo tanto, podemos asegurar que un lenguaje es regular si y solo si es aceptado por un AF.

Resumen

- Probamos que $ER^{\Sigma} \equiv AF^{\Sigma}$ (Teo. de Kleene)
- Por transitividad, tenemos toda la cadena de equivalencias probada:

$$LR^{\Sigma} \equiv ER^{\Sigma} \equiv AFD^{\Sigma} \equiv AFN^{\Sigma} \equiv AFN\epsilon^{\Sigma}$$

- Por lo tanto, podemos asegurar que un lenguaje es regular si y solo si es aceptado por un AF.
- En consecuencia, tenemos una nueva herramienta para probar regularidad de un lenguaje simplemente dando el autómata finito que lo acepta.

Resumen

- Probamos que $ER^{\Sigma} \equiv AF^{\Sigma}$ (Teo. de Kleene)
- Por transitividad, tenemos toda la cadena de equivalencias probada:

$$LR^{\Sigma} \equiv ER^{\Sigma} \equiv AFD^{\Sigma} \equiv AFN^{\Sigma} \equiv AFN\epsilon^{\Sigma}$$

- Por lo tanto, podemos asegurar que un lenguaje es regular si y solo si es aceptado por un AF.
- En consecuencia, tenemos una nueva herramienta para probar regularidad de un lenguaje simplemente dando el autómata finito que lo acepta.
- Podemos pensar este resultado como que los AF's son las máquinas secuenciales que caracterizan a los lenguajes regulares.

Resumen

- Probamos que $ER^{\Sigma} \equiv AF^{\Sigma}$ (Teo. de Kleene)
- Por transitividad, tenemos toda la cadena de equivalencias probada:

$$LR^{\Sigma} \equiv ER^{\Sigma} \equiv AFD^{\Sigma} \equiv AFN^{\Sigma} \equiv AFN\epsilon^{\Sigma}$$

- Por lo tanto, podemos asegurar que un lenguaje es regular si y solo si es aceptado por un AF.
- En consecuencia, tenemos una nueva herramienta para probar regularidad de un lenguaje simplemente dando el autómata finito que lo acepta.
- Podemos pensar este resultado como que los AF's son las máquinas secuenciales que caracterizan a los lenguajes regulares.
- En la próxima clase, presentaremos un nuevo modelo computacional que también caracteriza a los lenguajes regulares, las "gramáticas regulares".

Bibliografía

Rodrigo De Castro Korgi.
"Teoria de la Computación". Lenguajes, Autómatas, Gramáticas.
Capítulo 2: Autómatas Finitos. Sección 2.8 hasta 2.12.