

Autómatas y Lenguajes Formales 2016-1

Maestría en Ciencia e Ingeniería de la Computación UNAM

Tema 4: AFN con ε -transiciones, Teorema de Kleene I

Dr. Favio Ezequiel Miranda Perea
favio@ciencias.unam.mx

Facultad de Ciencias UNAM

29 de enero de 2020



AFN \Rightarrow AFD

Construcción de subconjuntos

Dado un AFN $M = \langle Q, \Sigma, \delta_N, q_0, F \rangle$ definimos un AFD $M^d = \langle Q^d, \Sigma^d, \delta, q_0^d, F^d \rangle$ como sigue:



AFN \Rightarrow AFD

Construcción de subconjuntos

Dado un AFN $M = \langle Q, \Sigma, \delta_N, q_0, F \rangle$ definimos un AFD $M^d = \langle Q^d, \Sigma^d, \delta, q_0^d, F^d \rangle$ como sigue:

- $Q^d = \mathcal{P}(Q)$.

AFN \Rightarrow AFD

Construcción de subconjuntos

Dado un AFN $M = \langle Q, \Sigma, \delta_N, q_0, F \rangle$ definimos un AFD $M^d = \langle Q^d, \Sigma^d, \delta, q_0^d, F^d \rangle$ como sigue:

- $Q^d = \mathcal{P}(Q)$.
- $\Sigma^d = \Sigma$.



AFN \Rightarrow AFD

Construcción de subconjuntos

Dado un AFN $M = \langle Q, \Sigma, \delta_N, q_0, F \rangle$ definimos un AFD $M^d = \langle Q^d, \Sigma^d, \delta, q_0^d, F^d \rangle$ como sigue:

- $Q^d = \mathcal{P}(Q)$.
- $\Sigma^d = \Sigma$.
- $\delta(S, a) = \bigcup_{q \in S} \delta_N(q, a)$



AFN \Rightarrow AFD

Construcción de subconjuntos

Dado un AFN $M = \langle Q, \Sigma, \delta_N, q_0, F \rangle$ definimos un AFD $M^d = \langle Q^d, \Sigma^d, \delta, q_0^d, F^d \rangle$ como sigue:

- $Q^d = \mathcal{P}(Q)$.
- $\Sigma^d = \Sigma$.
- $\delta(S, a) = \bigcup_{q \in S} \delta_N(q, a)$
- $q_0^d = \{q_0\}$



AFN \Rightarrow AFD

Construcción de subconjuntos

Dado un AFN $M = \langle Q, \Sigma, \delta_N, q_0, F \rangle$ definimos un AFD $M^d = \langle Q^d, \Sigma^d, \delta, q_0^d, F^d \rangle$ como sigue:

- $Q^d = \mathcal{P}(Q)$.
- $\Sigma^d = \Sigma$.
- $\delta(S, a) = \bigcup_{q \in S} \delta_N(q, a)$
- $q_0^d = \{q_0\}$
- $F^d = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$



AFN \Rightarrow AFD

Construcción de subconjuntos

Dado un AFN $M = \langle Q, \Sigma, \delta_N, q_0, F \rangle$ definimos un AFD $M^d = \langle Q^d, \Sigma^d, \delta, q_0^d, F^d \rangle$ como sigue:

- $Q^d = \mathcal{P}(Q)$.
- $\Sigma^d = \Sigma$.
- $\delta(S, a) = \bigcup_{q \in S} \delta_N(q, a)$
- $q_0^d = \{q_0\}$
- $F^d = \{S \subseteq Q \mid S \cap F \neq \emptyset\}$
- Ambos autómatas son equivalentes, es decir, $L(M) = L(M^d)$.



Autómatas No-deterministas con ε -transiciones

Definición

Un autómata finito **no** determinista con ε -transiciones (AFN ε) es una quintupla

$$M = \langle Q, \Sigma, \delta, q_0, F \rangle$$

Autómatas No-deterministas con ε -transiciones

Definición

Un autómata finito **no** determinista con ε -transiciones (AFN ε) es una quintupla

$$M = \langle Q, \Sigma, \delta, q_0, F \rangle$$

donde

- Q es un conjunto finito de estados.



Autómatas No-deterministas con ε -transiciones

Definición

Un autómata finito **no** determinista con ε -transiciones (AFN ε) es una quintupla

$$M = \langle Q, \Sigma, \delta, q_0, F \rangle$$

donde

- Q es un conjunto finito de estados.
- Σ es el alfabeto de entrada.



Autómatas No-deterministas con ε -transiciones

Definición

Un autómata finito **no** determinista con ε -transiciones (AFN ε) es una quintupla

$$M = \langle Q, \Sigma, \delta, q_0, F \rangle$$

donde

- Q es un conjunto finito de estados.
- Σ es el alfabeto de entrada.
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$ es la función de transición.



Autómatas No-deterministas con ε -transiciones

Definición

Un autómata finito **no** determinista con ε -transiciones (AFN ε) es una quintupla

$$M = \langle Q, \Sigma, \delta, q_0, F \rangle$$

donde

- Q es un conjunto finito de estados.
- Σ es el alfabeto de entrada.
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$ es la función de transición.
- $q_0 \in Q$ es el estado inicial.



Autómatas No-deterministas con ε -transiciones

Definición

Un autómata finito **no** determinista con ε -transiciones (AFN ε) es una quintupla

$$M = \langle Q, \Sigma, \delta, q_0, F \rangle$$

donde

- Q es un conjunto finito de estados.
- Σ es el alfabeto de entrada.
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$ es la función de transición.
- $q_0 \in Q$ es el estado inicial.
- $F \subseteq Q$ es el conjunto de estados finales.



AFN $_{\varepsilon}$

Observaciones

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$



AFN $_{\varepsilon}$

Observaciones

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$

- Obsérvese que la única diferencia está en el dominio de δ .



AFN $_{\varepsilon}$

Observaciones

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$

- Obsérvese que la única diferencia está en el dominio de δ .
- Transiciones de la forma $\delta(q, \varepsilon) = a$ están definidas.



AFN $_{\varepsilon}$

Observaciones

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$

- Obsérvese que la única diferencia está en el dominio de δ .
- Transiciones de la forma $\delta(q, \varepsilon) = a$ están definidas.
- Tales transiciones indican que la máquina puede cambiar de estado **sin** leer ningún símbolo.



AFN $_{\varepsilon}$

Observaciones

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$

- Obsérvese que la única diferencia está en el dominio de δ .
- Transiciones de la forma $\delta(q, \varepsilon) = a$ están definidas.
- Tales transiciones indican que la máquina puede cambiar de estado **sin** leer ningún símbolo.
- Esto causa un no-determinismo más complicado de modelar matemáticamente pero tiene grandes ventajas.



AFN ε

Observaciones

- Se permiten múltiples cómputos para una cadena de entrada.



AFN ε

Observaciones

- Se permiten múltiples cómputos para una cadena de entrada.
- Pueden existir cómputos bloqueados.



AFN ε

Observaciones

- Se permiten múltiples cómputos para una cadena de entrada.
- Pueden existir cómputos bloqueados.
- A diferencia de los AFD y AFN pueden existir cómputos infinitos, es decir, surge la no-terminación.



AFN ε

Observaciones

- Se permiten múltiples cómputos para una cadena de entrada.
- Pueden existir cómputos bloqueados.
- A diferencia de los AFD y AFN pueden existir cómputos infinitos, es decir, surge la no-terminación.
- La presencia de ε -transiciones permite mayor libertad en el diseño.



ε -cerradura de un estado

AFN ε

Dado un estado q , definimos la ε -cerradura de q como el conjunto de estados alcanzables desde q mediante cero o más ε -transiciones.



ε -cerradura de un estado

AFN ε

Dado un estado q , definimos la ε -cerradura de q como el conjunto de estados alcanzables desde q mediante cero o más ε -transiciones. Es decir

$$Cl_{\varepsilon}(q) = \{s \in Q \mid \exists p_1, \dots, p_n \text{ con } p_1 = q, p_n = s, p_i \in \delta(p_{i-1}, \varepsilon)\}$$

ε -cerradura de un estado

AFN ε

Dado un estado q , definimos la ε -cerradura de q como el conjunto de estados alcanzables desde q mediante cero o más ε -transiciones. Es decir

$$Cl_{\varepsilon}(q) = \{s \in Q \mid \exists p_1, \dots, p_n \text{ con } p_1 = q, p_n = s, p_i \in \delta(p_{i-1}, \varepsilon)\}$$

Tal definición se puede dar recursivamente:

- $q \in Cl_{\varepsilon}(q)$



ε -cerradura de un estado

AFN ε

Dado un estado q , definimos la ε -cerradura de q como el conjunto de estados alcanzables desde q mediante cero o más ε -transiciones. Es decir

$$Cl_{\varepsilon}(q) = \{s \in Q \mid \exists p_1, \dots, p_n \text{ con } p_1 = q, p_n = s, p_i \in \delta(p_{i-1}, \varepsilon)\}$$

Tal definición se puede dar recursivamente:

- $q \in Cl_{\varepsilon}(q)$
- Si $r \in Cl_{\varepsilon}(q)$ y $\delta(r, \varepsilon) = s$ entonces $s \in Cl_{\varepsilon}(q)$



ε -cerradura de un estado

AFN ε

Dado un estado q , definimos la ε -cerradura de q como el conjunto de estados alcanzables desde q mediante cero o más ε -transiciones. Es decir

$$Cl_{\varepsilon}(q) = \{s \in Q \mid \exists p_1, \dots, p_n \text{ con } p_1 = q, p_n = s, p_i \in \delta(p_{i-1}, \varepsilon)\}$$

Tal definición se puede dar recursivamente:

- $q \in Cl_{\varepsilon}(q)$
- Si $r \in Cl_{\varepsilon}(q)$ y $\delta(r, \varepsilon) = s$ entonces $s \in Cl_{\varepsilon}(q)$
- La definición de ε -cerradura se extiende a conjuntos de estados como sigue:

$$Cl_{\varepsilon}(S) = \bigcup_{q \in S} Cl_{\varepsilon}(q)$$



Función δ^*

AFN ε

Sea $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFN ε . La función de transición δ se extiende a cadenas mediante una función

$$\delta^* : Q \times (\Sigma^* \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$



Función δ^*

AFN ε

Sea $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFN ε . La función de transición δ se extiende a cadenas mediante una función

$$\delta^* : Q \times (\Sigma^* \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$

definida recursivamente como sigue:



Función δ^*

AFN ε

Sea $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFN ε . La función de transición δ se extiende a cadenas mediante una función

$$\delta^* : Q \times (\Sigma^* \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$

definida recursivamente como sigue:

$$\delta^*(q, \varepsilon) = Cl_\varepsilon(q)$$



Función δ^*

AFN ε

Sea $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ un AFN ε . La función de transición δ se extiende a cadenas mediante una función

$$\delta^* : Q \times (\Sigma^* \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$

definida recursivamente como sigue:

$$\delta^*(q, \varepsilon) = Cl_\varepsilon(q)$$

$$\delta^*(q, wa) = Cl_\varepsilon \left(\bigcup_{p \in \delta^*(q, w)} \delta(p, a) \right)$$



Eliminación de ε -transiciones

AFN \Rightarrow AFN $_{\varepsilon}$

Cualquier AFN $_{\varepsilon}$ es inmediatamente un AFN $_{\varepsilon}$ con la particularidad de que no existen ε -transiciones.



Eliminación de ε -transiciones

$\text{AFN}_\varepsilon \Rightarrow \text{AFN}$

Dado un $\text{AFN}_\varepsilon M_1 = \langle Q_1, \Sigma, \delta_1, q_{10}, F_1 \rangle$ existe un AFN equivalente $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, definido mediante:

- $Q := Q_1$

Eliminación de ε -transiciones

$\text{AFN}_\varepsilon \Rightarrow \text{AFN}$

Dado un $\text{AFN}_\varepsilon M_1 = \langle Q_1, \Sigma, \delta_1, q_{10}, F_1 \rangle$ existe un AFN equivalente $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, definido mediante:

- $Q := Q_1$
- $q_0 := q_{10}$



Eliminación de ε -transiciones

AFN $\varepsilon \Rightarrow$ AFN

Dado un AFN ε $M_1 = \langle Q_1, \Sigma, \delta_1, q_{10}, F_1 \rangle$ existe un AFN equivalente $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, definido mediante:

- $Q := Q_1$
- $q_0 := q_{10}$
- $\delta(q, a) = \delta_1^*(q, a) = CI_\varepsilon \left(\bigcup_{p \in CI_\varepsilon(q)} \delta_1(p, a) \right)$

Eliminación de ε -transiciones

$\text{AFN}_\varepsilon \Rightarrow \text{AFN}$

Dado un $\text{AFN}_\varepsilon M_1 = \langle Q_1, \Sigma, \delta_1, q_{10}, F_1 \rangle$ existe un AFN equivalente $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, definido mediante:

- $Q := Q_1$
- $q_0 := q_{10}$
- $\delta(q, a) = \delta_1^*(q, a) = CI_\varepsilon \left(\bigcup_{p \in CI_\varepsilon(q)} \delta_1(p, a) \right)$
- $F := F_1 \cup \{q_{10}\}$ si $CI_\varepsilon(q_{10}) \cap F_1 \neq \emptyset$



Eliminación de ε -transiciones

$AFN_{\varepsilon} \Rightarrow AFN$

Dado un AFN_{ε} $M_1 = \langle Q_1, \Sigma, \delta_1, q_{10}, F_1 \rangle$ existe un AFN equivalente $M = \langle Q, \Sigma, \delta, q_0, F \rangle$, definido mediante:

- $Q := Q_1$
- $q_0 := q_{10}$
- $\delta(q, a) = \delta_1^*(q, a) = CI_{\varepsilon}\left(\bigcup_{p \in CI_{\varepsilon}(q)} \delta_1(p, a)\right)$
- $F := F_1 \cup \{q_{10}\}$ si $CI_{\varepsilon}(q_{10}) \cap F_1 \neq \emptyset$
- $F := F_1$ en caso contrario.



Autómatas Finitos

AFD \Leftrightarrow AFN \Leftrightarrow AFN $_{\varepsilon}$

- Con la equivalencia AFN \Leftrightarrow AFN $_{\varepsilon}$ hemos cerrado el ciclo de equivalencias de autómatas finitos.



Autómatas Finitos

AFD \Leftrightarrow AFN \Leftrightarrow AFN $_{\varepsilon}$

- Con la equivalencia AFN \Leftrightarrow AFN $_{\varepsilon}$ hemos cerrado el ciclo de equivalencias de autómatas finitos.
- Cualquier tipo de autómata finito puede convertirse en un AFD y viceversa, es decir:

$$AFD \Leftrightarrow AFN \Leftrightarrow AFN_{\varepsilon}$$

Autómatas Finitos

AFD \Leftrightarrow AFN \Leftrightarrow AFN $_{\varepsilon}$

- Con la equivalencia AFN \Leftrightarrow AFN $_{\varepsilon}$ hemos cerrado el ciclo de equivalencias de autómatas finitos.
- Cualquier tipo de autómata finito puede convertirse en un AFD y viceversa, es decir:

$$AFD \Leftrightarrow AFN \Leftrightarrow AFN_{\varepsilon}$$

- Nuestra siguiente meta es probar el teorema de Kleene, el cual es uno de los resultados más importantes en la teoría de la computación pues asegura la equivalencia entre dos de nuestros tres conceptos fundamentales, los autómatas finitos y los lenguajes regulares.



Teorema de Kleene

Teorema (Kleene)

Un lenguaje es regular si y sólo si es aceptado por un autómata finito.



Teorema de Kleene

Teorema (Kleene)

Un lenguaje es regular si y sólo si es aceptado por un autómata finito.

Demostración.

La prueba es en dos partes:



Teorema de Kleene

Teorema (Kleene)

Un lenguaje es regular si y sólo si es aceptado por un autómata finito.

Demostración.

La prueba es en dos partes:

- I Síntesis: Dado un lenguaje regular L existe un autómata finito M tal que $L = L(M)$.



Teorema de Kleene

Teorema (Kleene)

Un lenguaje es regular si y sólo si es aceptado por un autómata finito.

Demostración.

La prueba es en dos partes:

- I Síntesis: Dado un lenguaje regular L existe un autómata finito M tal que $L = L(M)$.
- II Análisis: Dado un autómata finito M existe una expresión regular α tal que $L(M) = L(\alpha)$. Es decir, $L(M)$ es regular.



Teorema de Kleene I

ER \Rightarrow AF

- El teorema de síntesis de Kleene se probará construyendo un autómata finito para un lenguaje regular dado analizando la forma de la expresión regular que genera al lenguaje.



Teorema de Kleene I

ER \Rightarrow AF

- El teorema de síntesis de Kleene se probará construyendo un autómata finito para un lenguaje regular dado analizando la forma de la expresión regular que genera al lenguaje.
- La demostración es constructiva y se hará mediante inducción sobre las expresiones regulares.



Teorema de Kleene I

Base de la Inducción, $\alpha = \emptyset$

Si $\alpha = \emptyset$ entonces el siguiente autómata reconoce a $L(\alpha)$:



Teorema de Kleene I

Base de la Inducción, $\alpha = \varepsilon$

Si $\alpha = \varepsilon$ entonces el siguiente autómata reconoce a $L(\alpha)$:



Teorema de Kleene I

Base de la Inducción, $\alpha = a$

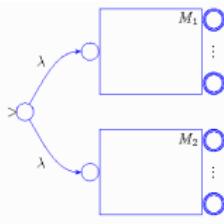
Si $\alpha = a$ entonces el siguiente autómata reconoce a $L(\alpha)$:



Teorema de Kleene I

Paso Inductivo, $\alpha = \alpha_1 + \alpha_2$

Si $\alpha = \alpha_1 + \alpha_2$ entonces el siguiente autómata reconoce a $L(\alpha)$:



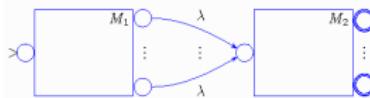
donde M_1, M_2 son autómatas que reconocen a $L(\alpha_1), L(\alpha_2)$ dados por la hipótesis de inducción.



Teorema de Kleene I

Paso Inductivo, $\alpha = \alpha_1\alpha_2$

Si $\alpha = \alpha_1\alpha_2$ entonces el siguiente autómata reconoce a $L(\alpha)$:



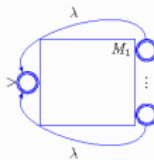
donde M_1, M_2 son autómatas que reconocen a $L(\alpha_1), L(\alpha_2)$ dados por la hipótesis de inducción.



Teorema de Kleene I

Paso Inductivo, $\alpha = \alpha_1^*$

Si $\alpha = \alpha_1^*$ entonces el siguiente autómata reconoce a $L(\alpha)$:



donde M_1 es un autómata que reconoce a $L(\alpha_1)$ dados por la hipótesis de inducción.



Teorema de Análisis de Kleene

$AF \Rightarrow ER$

Teorema (Análisis de Kleene)

Dado un autómata finito M existe una expresión regular α tal que $L(M) = L(\alpha)$. Es decir, $L(M)$ es regular.

Teorema de Análisis de Kleene

$AF \Rightarrow ER$

Teorema (Análisis de Kleene)

Dado un autómata finito M existe una expresión regular α tal que $L(M) = L(\alpha)$. Es decir, $L(M)$ es regular.

Demostración.

Existen diversas demostraciones, nosotros usaremos el método de ecuaciones características usando el lema de Arden.



Ecuaciones de lenguajes

Kleene II

Sean $A, B \subseteq \Sigma^*$ y X una variable:

- Una ecuación lineal derecha para X es una expresión de la forma:

$$X = AX + B$$



Ecuaciones de lenguajes

Kleene II

Sean $A, B \subseteq \Sigma^*$ y X una variable:

- Una ecuación lineal derecha para X es una expresión de la forma:

$$X = AX + B$$

- Análogamente, una ecuación lineal izquierda es una expresión de la forma:

$$X = XA + B$$



Ecuaciones de lenguajes

Kleene II

Sean $A, B \subseteq \Sigma^*$ y X una variable:

- Una ecuación lineal derecha para X es una expresión de la forma:

$$X = AX + B$$

- Análogamente, una ecuación lineal izquierda es una expresión de la forma:

$$X = XA + B$$

- Donde el símbolo $+$ denota a la unión de lenguajes.



Lema de Arden

Sean $A, B \subseteq \Sigma^*$ dos lenguajes y $X = AX + B$ una ecuación lineal derecha.

Lema de Arden

Sean $A, B \subseteq \Sigma^*$ dos lenguajes y $X = AX + B$ una ecuación lineal derecha. Entonces

- A^*B es una solución de la ecuación, es decir, $A^*B = A(A^*B) + B$.



Lema de Arden

Sean $A, B \subseteq \Sigma^*$ dos lenguajes y $X = AX + B$ una ecuación lineal derecha. Entonces

- A^*B es una solución de la ecuación, es decir, $A^*B = A(A^*B) + B$.
- Si C es otra solución entonces $A^*B \subseteq C$, es decir, A^*B es la solución mínima.



Lema de Arden

Sean $A, B \subseteq \Sigma^*$ dos lenguajes y $X = AX + B$ una ecuación lineal derecha. Entonces

- A^*B es una solución de la ecuación, es decir, $A^*B = A(A^*B) + B$.
- Si C es otra solución entonces $A^*B \subseteq C$, es decir, A^*B es la solución mínima.
- Si $\varepsilon \notin A$ entonces A^*B es la única solución.



El sistema de ecuaciones de un AFN

$AF \Rightarrow ER$

Dado un AFN $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ tal que $Q = \{q_0, \dots, q_n\}$. Definimos los siguientes conjuntos:



El sistema de ecuaciones de un AFN

$AF \Rightarrow ER$

Dado un AFN $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ tal que $Q = \{q_0, \dots, q_n\}$. Definimos los siguientes conjuntos:

- El conjunto de cadenas que se aceptan desde el estado q_i , para cualquier $1 \leq i \leq n$:

$$L_i = \{w \in \Sigma^* \mid \delta^*(q_i, w) \cap F \neq \emptyset\}$$



El sistema de ecuaciones de un AFN

$AF \Rightarrow ER$

Dado un AFN $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ tal que $Q = \{q_0, \dots, q_n\}$. Definimos los siguientes conjuntos:

- El conjunto de cadenas que se aceptan desde el estado q_i , para cualquier $1 \leq i \leq n$:

$$L_i = \{w \in \Sigma^* \mid \delta^*(q_i, w) \cap F \neq \emptyset\}$$

- L_0 es el lenguaje aceptado por M , es decir, $L_0 = L(M)$.



El sistema de ecuaciones de un AFN

$AF \Rightarrow ER$

Dado un AFN $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ tal que $Q = \{q_0, \dots, q_n\}$. Definimos los siguientes conjuntos:

- El conjunto de cadenas que se aceptan desde el estado q_i , para cualquier $1 \leq i \leq n$:

$$L_i = \{w \in \Sigma^* \mid \delta^*(q_i, w) \cap F \neq \emptyset\}$$

- L_0 es el lenguaje aceptado por M , es decir, $L_0 = L(M)$.
- En general no es sencillo calcular directamente los conjuntos L_i .



El sistema de ecuaciones de un AFN

$AF \Rightarrow ER$

- El conjunto de símbolos de Σ tal que existe una transición del estado q_i al estado q_j , para cualesquiera $1 \leq i, j \leq n$.

$$X_{i,j} = \{a \in \Sigma \mid q_j \in \delta(q_i, a)\}$$

El sistema de ecuaciones de un AFN

$AF \Rightarrow ER$

- El conjunto de símbolos de Σ tal que existe una transición del estado q_i al estado q_j , para cualesquiera $1 \leq i, j \leq n$.

$$X_{i,j} = \{a \in \Sigma \mid q_j \in \delta(q_i, a)\}$$

- El conjunto auxiliar Y_i que indica si ε es aceptada desde q_i

$$Y_i = \begin{cases} \{\varepsilon\} & \text{si } q_i \in F \\ \emptyset & \text{en otro caso} \end{cases}$$



El sistema de ecuaciones de un AFN

AF \Rightarrow ER

La siguiente propiedad es fácil de demostrar para cualquier $1 \leq i \leq n$.

$$L_i = \sum_{j=0}^n X_{i,j} L_j + Y_i$$

El sistema de ecuaciones de un AFN

AF \Rightarrow ER

La siguiente propiedad es fácil de demostrar para cualquier $1 \leq i \leq n$.

$$L_i = \sum_{j=0}^n X_{i,j} L_j + Y_i$$

- Dicha propiedad genera el llamado sistema de ecuaciones características de una AFN.



El sistema de ecuaciones de un AFN

AF \Rightarrow ER

La siguiente propiedad es fácil de demostrar para cualquier $1 \leq i \leq n$.

$$L_i = \sum_{j=0}^n X_{i,j} L_j + Y_i$$

- Dicha propiedad genera el llamado sistema de ecuaciones características de una AFN.
- El lema de Arden nos indica como calcular los conjuntos L_i .



Teorema de Análisis de Kleene

Idea de la demostración

- Dado el autómata M construir los conjuntos

$$X_{i,j}, \quad Y_i$$



Teorema de Análisis de Kleene

Idea de la demostración

- Dado el autómata M construir los conjuntos

$$X_{i,j}, \ Y_i$$

- Resolver el sistema de ecuaciones características mediante el Lema de Arden.



Teorema de Análisis de Kleene

Idea de la demostración

- Dado el autómata M construir los conjuntos

$$X_{i,j}, \quad Y_i$$

- Resolver el sistema de ecuaciones características mediante el Lema de Arden.
- La solución para L_0 genera una expresión regular para $L(M)$.

