

1.6. Equivalencia entre AFND y AFD

Teorema. Todo AFND tiene un AFD equivalente.

Corolario. Sea $L \subseteq \Sigma^*$ un lenguaje. El lenguaje L es regular si y sólo si existe un AFND M tal que

$$\mathcal{L}(M) = L.$$

1.7. Operaciones regulares

Teorema. Sean $L_1, L_2 \subseteq \Sigma^*$ lenguajes regulares. Los siguientes lenguajes son regulares:

$$L_1 \cup L_2, \quad L_1 L_2, \quad L_1^*.$$

Observación. Todo lenguaje finito es regular.

1.8. Expresiones regulares

Definición. Las *expresiones regulares sobre un alfabeto Σ* forman el menor conjunto de expresiones que satisface las siguientes reglas:

- cada símbolo $\sigma \in \Sigma$ es una expresión regular;
- $\emptyset$ y ε son expresiones regulares;
- si R_1 y R_2 son expresiones regulares sobre Σ , entonces

$$(R_1 \mid R_2), \quad (R_1 R_2), \quad (R_1^*)$$

son expresiones regulares sobre Σ .

Definición. Sea R una expresión regular sobre Σ . El *lenguaje descrito por R* , denotado por $\mathcal{L}(R)$, se define recursivamente por

$$\begin{aligned} \mathcal{L}(\sigma) &= \{\sigma\} && \text{para cada } \sigma \in \Sigma, \\ \mathcal{L}(\emptyset) &= \emptyset, \\ \mathcal{L}(\varepsilon) &= \{\varepsilon\}, \\ \mathcal{L}(R_1 \mid R_2) &= \mathcal{L}(R_1) \cup \mathcal{L}(R_2), \\ \mathcal{L}(R_1 R_2) &= \mathcal{L}(R_1) \mathcal{L}(R_2), \\ \mathcal{L}(R_1^*) &= \mathcal{L}(R_1)^*. \end{aligned}$$

Teorema. Sea $L \subseteq \Sigma^*$ un lenguaje. El lenguaje L es regular si y sólo si existe una expresión regular R sobre Σ tal que $\mathcal{L}(R) = L$.

Ejercicios

1. a) Considere el AFND $N = (Q, \Sigma, \delta, s, F)$, donde $Q = \{s, p, q\}$, $\Sigma = \{0, 1\}$, $F = \{q\}$ y

	0	1
s	$\{s, p\}$	$\{s\}$
p	$\emptyset$	$\{q\}$
q	$\emptyset$	$\emptyset$

Construya un AFD y una ER que acepten el mismo lenguaje que N .

- b) Para cada una de las siguientes expresiones regulares sobre $\Sigma = \{0, 1\}$, describa su lenguaje

$$(00 \mid 11)^*, \quad 0^*(10^*10^*)^*, \quad (0 \mid 10)^*(\varepsilon \mid 1).$$

Pruebe que una de sus descripciones es correcta via inducción estructural.

- c) Entregue una expresión regular para cada uno de los siguientes lenguajes sobre $\Sigma = \{0, 1\}$:
- 1) los strings que contienen exactamente dos unos y al menos un cero;
 - 2) los strings cuyo largo es 1 módulo 4;
 - 3) los strings en que cada cero es seguido por al menos tres unos.
- d) Sean $A, B \subseteq \Sigma^*$ lenguajes regulares y sea $k \geq 0$ una constante. Demuestre que los siguientes son regulares

$$A \setminus B, \quad A^+ = AA^*, \quad \bigcup_{i=0}^k A^i.$$

- e) ¿Qué clase de lenguajes describen las expresiones regulares que no contienen el símbolo $*$?

2. La siguiente pregunta modela el daño que puede sufrir un mensaje durante su transmisión: algunos caracteres pueden perderse, pero el orden de los caracteres que llegan es el mismo que el del mensaje original. Diremos que x es una *versión dañada* de w si x puede obtenerse borrando cero o más símbolos de w . Para un lenguaje $L \subseteq \Sigma^*$, definimos

$$\text{Daño}(L) = \{x \in \Sigma^* : x \text{ es una versión dañada de algún } w \in L\}.$$

- a) Un protocolo acepta los mensajes binarios descritos por

$$R_3 = 0^*10^*10^*10^*.$$

Entregue una expresión regular que describa todas las versiones dañadas de los mensajes aceptados por el protocolo.

- b) Sea R una expresión regular cualquiera. Defina recursivamente una expresión regular $D(R)$ que describa todas las versiones dañadas de los strings descritos por R .
c) Demuestre por inducción estructural sobre R que su construcción satisface

$$\mathcal{L}(D(R)) = \text{Daño}(\mathcal{L}(R)).$$

Deduzca que, si L es regular, entonces $\text{Daño}(L)$ también es regular.

- d) Demuestre además que, para todos $L, K \subseteq \Sigma^*$ tenemos que $L \subseteq \text{Daño}(L)$, si $L \subseteq K$ entonces $\text{Daño}(L) \subseteq \text{Daño}(K)$. Concluya que $\text{Daño}(\text{Daño}(L)) = \text{Daño}(L)$.
e) Considere el siguiente problema computacional: dados una expresión regular R y un string recibido x , decidir si x podría ser una versión dañada de algún mensaje de $\mathcal{L}(R)$. Proponga un algoritmo eficiente para resolver este problema y discuta sobre su análisis. ¿Que pasa si el problema busca recuperar el mensaje original w a partir de x y R ?

3. Sea $L \subseteq \Sigma^*$ un lenguaje y sea $a \in \Sigma$. El *cociente izquierdo de L por a* es

$$a^{-1}L = \{x \in \Sigma^* : ax \in L\}.$$

Para una expresión regular R , definimos

$$\text{nul}(R) = \begin{cases} \varepsilon, & \text{si } \varepsilon \in \mathcal{L}(R), \\ \emptyset, & \text{si } \varepsilon \notin \mathcal{L}(R). \end{cases}$$

La *derivada de R respecto de a* , denotada por $\partial_a R$, se define recursivamente por

$$\begin{aligned} \partial_a \emptyset &= \emptyset, \\ \partial_a \varepsilon &= \emptyset, \\ \partial_a b &= \begin{cases} \varepsilon, & \text{si } a = b, \\ \emptyset, & \text{si } a \neq b, \end{cases} & \text{para } b \in \Sigma, \\ \partial_a (R \mid S) &= (\partial_a R) \mid (\partial_a S), \\ \partial_a (RS) &= (\partial_a R)S \mid \text{nul}(R)(\partial_a S), \\ \partial_a (R^*) &= (\partial_a R)R^*. \end{aligned}$$

- a) Demuestre por inducción estructural que $\mathcal{L}(\partial_a R) = a^{-1}\mathcal{L}(R)$.
- b) Extienda la derivada a strings mediante $\partial_\varepsilon R = R$ y $\partial_{wa} R = \partial_a(\partial_w R)$. Demuestre que, para todo $w \in \Sigma^*$,

$$\mathcal{L}(\partial_w R) = \{z \in \Sigma^* : wz \in \mathcal{L}(R)\},$$

y deduzca un criterio para decidir si $w \in \mathcal{L}(R)$ mirando solamente si $\partial_w R$ acepta a ε .

- c) Considere $R = (0 \mid 1)^*001$. Calcule todas sus derivadas alcanzables, identificando dos expresiones cuando describen el mismo lenguaje.
- d) Diremos que dos strings $u, v \in \Sigma^*$ son *indistinguibles para L* , y escribiremos $u \equiv_L v$, si

$$uz \in L \iff vz \in L \quad \text{para todo } z \in \Sigma^*.$$

Pruebe que $\equiv_L$ es una relación de equivalencia y que

$$u \equiv_{\mathcal{L}(R)} v \iff \mathcal{L}(\partial_u R) = \mathcal{L}(\partial_v R).$$

Si un AFD M reconoce L , demuestre que dos strings que llevan a M al mismo estado son indistinguibles para L . Concluya que cualquier AFD que reconoce L tiene al menos tantos estados como clases de equivalencia de $\equiv_L$.

- e) Suponga ahora que $\equiv_L$ tiene una cantidad finita de clases. Construya un AFD con un estado $[u]$ por cada clase, estado inicial $[\varepsilon]$, estados finales $\{[u] : u \in L\}$ y transiciones $[u] \xrightarrow{a} [ua]$. Pruebe que la construcción está bien definida y reconoce L .
- f) Concluya el *teorema de Myhill–Nerode*: un lenguaje es regular si y sólo si $\equiv_L$ tiene finitas clases; además, el número de clases es el número de estados de un AFD mínimo.