Introducción a la Teoría de la Computación
I Semestre 2023
Profesor: Rodrigo De Castro K.

Capítulo 1

Alfabetos, cadenas y lenguajes

De manera muy amplia podría decirse que la computación es la manipulación de entradas (*inputs*) para producir salidas (*outputs*). Pero cualquier mecanismo de cómputo solamente puede manipular un número finito de símbolos en un tiempo finito. Desde el punto de vista teórico esto impone dos restricciones naturales: el conjunto de símbolos (alfabeto) debe ser finito y se deben considerar únicamente cadenas (secuencias de símbolos) de longitud finita. Surgen así los ingredientes esenciales de una teoría abstracta de la computación: *alfabetos* y *cadenas*. Los conjuntos de cadenas (ya sean finitos o infinitos) se denominarán *lenguajes*.

1.1. Alfabetos y cadenas

Alfabetos. Un *alfabeto* es un conjunto finito no vacío cuyos elementos se llaman *símbolos*. Es costumbre usar las letras griegas mayúsculas Σ, Γ, Δ para denotar alfabetos. De ser necesario se utilizan también subíndices: $\Sigma_1, \Sigma_2, \Sigma_3, \ldots$.

Un alfabeto genérico Σ consta de k símbolos diferentes, $s_1, s_2, \ldots, s_k$, donde $k \geq 1$; esto es, $\Sigma = \{s_1, s_2, \ldots, s_k\}$. A continuación presentamos algunos de los alfabetos más conocidos.

Ejemplos

- El alfabeto latino (letras minúsculas, 26 símbolos).

  $\Sigma_1 = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z\}$.

- El alfabeto latino (letras mayúsculas, 26 símbolos).

  $\Sigma_2 = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z\}$.

- El alfabeto latino (letras minúsculas y mayúsculas, 52 símbolos).

  $\Sigma_3 = \{a, A, b, B, c, C, d, D, e, E, \ldots, w, W, x, X, y, Y, z, Z\}$.

- El alfabeto del idioma español o castellano (letras minúsculas) consta de los 26 símbolos del alfabeto latino más los símbolos acentuados *á, é, í, ó, ú, ñ*. Esto es,

  $\Sigma_4 = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \text{á}, \text{é}, \text{í}, \text{ó}, \text{ú}, \text{ñ}\}$.

- El alfabeto griego (letras minúsculas, 24 símbolos).

  $\Sigma_5 = \{\alpha, \beta, \gamma, \delta, \epsilon, \eta, \iota, \kappa, \lambda, \mu, \nu, o, \psi, \phi, \pi, \rho, \sigma, \tau, \theta, \upsilon, \chi, \xi, \zeta, \omega\}$.

- El alfabeto de los dígitos (10 símbolos).

  $\Sigma_6 = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 0\}$.

- El alfabeto $\Sigma = \{0, 1\}$ se conoce como *alfabeto binario*.

El alfabeto binario $\Sigma = \{0, 1\}$ se podría considerar como el más importante ya que cualquier otro alfabeto se puede codificar usando solamente ceros y unos. La noción de *codificación* no es relevante por el momento y se estudiará en detalle en el capítulo 6.

Cadenas. Una *cadena* o *palabra* sobre un alfabeto Σ dado es cualquier sucesión (o secuencia) finita de elementos de Σ. Según las convenciones estándares, una cadena se escribe de izquierda a derecha. Admitimos la existencia de una única cadena que no tiene símbolos, la cual se denomina *cadena vacía* y se denota con λ. La cadena vacía desempeña, en la teoría de la computación, un papel similar al del conjunto vacío $\varnothing$ en la teoría de conjuntos.

Ejemplo Sea $\Sigma = \{a, b\}$ el alfabeto que consta de los dos símbolos a y b. Las siguientes son cadenas sobre Σ:

aba
$ababaaa$
$aaaab$.

Obsérvese que $aba \neq aab$ y que $aaab \neq baaa$. En una cadena hay que tener en cuenta el orden en el que aparecen los símbolos ya que las cadenas se definen como *sucesiones*, es decir, conjuntos *secuencialmente ordenados*.

Ejemplo Las cadenas sobre el alfabeto binario $\Sigma = \{0, 1\}$ son secuencias finitas de ceros y unos, llamadas *secuencias binarias*, tales como

001
1011
001000001.

En general, dado un alfabeto de referencia Σ, una cadena genérica es una secuencia de la forma $a_1 a_2 \cdots a_n$, donde $a_1, a_2, \ldots, a_n \in \Sigma$. Para denotar cadenas concretas se utilizan letras como u, v, w, x, y, z, con subíndices si es necesario; si tales letras hacen parte del alfabeto Σ, entonces se utilizan letras griegas o símbolos completamente diferentes a los de Σ.

Ejemplo Sea $\Sigma = \{a, b, c, d\}$. Utilizamos u, v, w para denotar (dar un nombre) a ciertas cadenas concretas:

$u = ddaba.$
$v = cababadda.$
$w = bbbcc.$

Longitud de una cadena. La *longitud* de una cadena $u \in \Sigma^*$ se denota $|u|$ y se puede definir descriptivamente como el número de símbolos de u (contando los símbolos repetidos). Es decir,

$$|u| = \begin{cases} 0, & \text{si } u = \lambda, \\ n, & \text{si } u = a_1 a_2 \cdots a_n, \text{donde } a_1, a_2, \ldots, a_n \in \Sigma \ y \ n \geq 1. \end{cases}$$

Ejemplo Sea $\Sigma = \{a, b, c, d\}$. Si $u = bbcada \Longrightarrow |u| = 6$. Si $v = ccddd \Longrightarrow |v| = 5$.

Se puede observar que los símbolos de un alfabeto Σ juegan un doble papel: como símbolos del alfabeto de referencia y como cadenas de longitud uno.

Definición descriptiva de Σ^*. El conjunto de *todas* las cadenas sobre un alfabeto Σ, incluyendo la cadena vacía, se denota por Σ^*.

Ejemplo Sea $\Sigma = \{a, b, c\}$, entonces

$$\Sigma^* = \{\lambda, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, aba, \ldots\}.$$

En este caso, Σ^* está formado por la cadena vacía λ, las 3 cadenas de longitud uno, a, b, c; las 9 cadenas de longitud dos, $aa, ab, ac, ba, bb, bc, ca, cb, cc$; las 27 cadenas de longitud tres, $aaa, aab, abc, baa, \ldots, ccc$, etc.

✎ Algunos autores denotan la cadena vacía con la letra griega ε. Preferimos denotarla con λ porque ε tiende a confundirse con el símbolo $\in$ usado para la relación de pertenencia. También es frecuente usar Λ para denotar la cadena vacía.

✎ Si bien un alfabeto Σ es un conjunto finito, Σ^* es siempre un conjunto infinito (enumerable). En el caso más simple, Σ contiene solo un símbolo, por ejemplo, $\Sigma = \{a\}$, y $\Sigma^* = \{\lambda, a, aa, aaa, aaaa, aaaaa, \ldots\}$.

✎ $\varnothing$ y λ son objetos diferentes: $\varnothing$ es un conjunto (el único conjunto que no tiene elementos) y λ es una cadena (la única cadena que no tiene símbolos).

✎ La mayor parte de la teoría de la computación se hace con referencia a un alfabeto Σ fijo (pero arbitrario).

1.2. Definiciones y demostraciones recursivas

En la Teoría de la Computación muchos conjuntos se definen recursivamente. A continuación presentamos este tipo de definiciones en un contexto muy amplio.

Definición recursiva de un conjunto. Un conjunto S se define recursivamente por medio de tres cláusulas:

(1) Cláusula básica. Se especifican los elementos más sencillos (o básicos) de S.

(2) Cláusula recursiva. Se especifican la regla o reglas para la formación de nuevos elementos de S a partir de elementos ya conocidos de S.

(3) Cláusula de exclusión. Establece que los elementos de S se pueden obtener únicamente utilizando las cláusulas (1) y (2).

La cláusula de exclusión se suele admitir de manera implícita y es corriente omitirla en las definiciones recursivas concretas, algo que haremos en lo sucesivo.

Ejemplo Dado un alfabeto Σ, el conjunto Σ^* de todas las cadenas, definido descriptivamente en la sección 1.1, se puede definir recursivamente. La idea es que a partir de una cadena $u \in \Sigma^*$ se puede obtener una nueva cadena añadiendo a la derecha un símbolo a perteneciente al alfabeto Σ; dicha cadena será denotada por ua. Todas las cadenas de Σ^* se pueden obtener de esta manera a partir de la cadena vacía λ; se entiende que $\lambda a = a$, para cualquier símbolo $a \in \Sigma$.

La definición recursiva de Σ^* consta entonces de las siguientes dos cláusulas:

(1) $\lambda \in \Sigma^*$.

(2) Si $u \in \Sigma^*$ entonces $ua \in \Sigma^*$ para cada símbolo $a \in \Sigma$, donde ua es la secuencia de símbolos obtenida a partir de u añadiendo el símbolo a a la derecha.

Definición recursiva de conceptos sobre cadenas. La definición recursiva de Σ^* permite definir recursivamente nociones o conceptos aplicables a todas las cadenas. Sea Σ un alfabeto dado; para definir un concepto C sobre todas las cadenas en Σ^* se utilizan dos cláusulas:

(1) Se define $C(\lambda)$, es decir, se define el concepto C para la cadena λ.

(2) Se define $C(ua)$ en términos de $C(u)$ para toda cadena $u \in \Sigma^*$ y todo símbolo $a \in \Sigma$.

Ejemplo Aunque la noción de longitud de una cadena es muy simple y bastaría la descripción presentada en la sección 1.1, también podemos dar una definición recursiva de $|u|$, con $u \in \Sigma^*$:

(1) $|\lambda| = 0$.

(2) $|ua| = |u| + 1$ para toda $u \in \Sigma^*$ y todo símbolo $a \in \Sigma$.

Recursión sobre cadenas. Dado un alfabeto Σ, es posible demostrar recursivamente que todas las cadenas en Σ^* satisfacen una cierta propiedad P. A tal tipo de razonamiento recursivo se le conoce como *recursión sobre cadenas*, y consta de dos pasos:

(1) Se demuestra $P(\lambda)$, es decir, se demuestra que la cadena vacía λ satisface la propiedad P.

(2) Se demuestra la implicación

$$P(u) \Longrightarrow P(ua), \text{ para toda } u \in \Sigma^* \text{ y todo } a \in \Sigma.$$

Es decir, se demuestra que si u satisface la propiedad P, entonces la cadena ua también satisface la propiedad P, para todo $a \in \Sigma$. $P(u)$ es la llamada *hipótesis recursiva*, a partir de la cual se demuestra $P(ua)$ para todo $a \in \Sigma$.

La recursión sobre cadenas es un razonamiento demostrativo similar a la llamada *inducción matemática* que se utiliza para demostrar propiedades sobre los números naturales. Estos argumentos son necesarios cuando se quiere demostrar propiedades con toda rigurosidad.

1.3. Concatenación de cadenas

Dado un alfabeto Σ y dos cadenas $u, v \in \Sigma^*$, la *concatenación de u y v* se denota como $u \cdot v$ o simplemente uv y se define descriptivamente así:

(1) Si $v = \lambda$, entonces $u \cdot \lambda = \lambda \cdot u = u$. Es decir, la concatenación de cualquier cadena u con la cadena vacía, a izquierda o a derecha, es igual a u.

(2) Si $u = a_1a_2 \cdots a_n$, $v = b_1b_2 \cdots b_m$, donde $a_1, \ldots, a_n, b_1, \ldots, b_m \in \Sigma$, $n, m \geq 1$, entonces

$$u \cdot v = a_1a_2 \cdots a_nb_1b_2 \cdots b_m.$$

Es decir, $u \cdot v$ es la cadena formada escribiendo los símbolos de u y a continuación los símbolos de v.

Según la definición, se tiene que $|u \cdot v| = |u| + |v|$. En general, $u \cdot v \neq v \cdot u$, es decir, la concatenación no es una operación conmutativa.

Ejemplo Sea $\Sigma = \{a, b, c, d\}$. Si $u = bcaad$ y $v = dcbb$, entonces $u \cdot v = bcaaddcbb$ y $v \cdot u = dcbbbcaad$.

Definición recursiva de la concatenación. Para definir recursivamente la concatenación $u \cdot v$ para todas las cadenas $u, v \in \Sigma^*$ se toma una cadena fija u (pero arbitraria) y se hace recursión sobre v, de la siguiente manera:

(1) $u \cdot \lambda = \lambda \cdot u = u$.

(2) $u \cdot (va) = (u \cdot v)a$, para toda $v \in \Sigma^*$, $a \in \Sigma$.

Propiedad asociativa de la concatenación. La concatenación de cadenas es una operación asociativa. Es decir, si $u, v, w \in \Sigma^*$, entonces

$$(u \cdot v) \cdot w = u \cdot (v \cdot w).$$

Demostración. Primero hay que considerar los casos en los que alguna (o algunas) de las cadenas u, v o w sea la cadena vacía λ. Por ejemplo, si $v = \lambda$ tenemos $(u \cdot v) \cdot w = (u \cdot \lambda) \cdot w = u \cdot w$ mientras que $u \cdot (v \cdot w) = u \cdot (\lambda \cdot w) = u \cdot w$.

Luego consideramos el caso en que u, v y w sean diferentes de λ. Entonces podemos escribir

$$\begin{aligned} u &= a_1 \cdots a_n, \text{ con } a_i \in \Sigma, n \geq 1. \\ v &= b_1 \cdots b_m, \text{ con } b_i \in \Sigma, m \geq 1. \\ w &= c_1 \cdots c_k, \text{ con } c_i \in \Sigma, k \geq 1. \end{aligned}$$

Entonces

$$\begin{aligned} (u \cdot v) \cdot w &= (a_1 \cdots a_n b_1 \cdots b_m) \cdot w = a_1 \cdots a_n b_1 \cdots b_m c_1 \cdots c_k. \\ u \cdot (v \cdot w) &= u \cdot (b_1 \cdots b_m c_1 \cdots c_k) = a_1 \cdots a_n b_1 \cdots b_m c_1 \cdots c_k. \end{aligned}$$

Esto demuestra la propiedad asociativa.

La propiedad asociativa también se puede demostrar por recursión sobre cadenas: se mantienen u y v fijas (pero arbitrarias) y se hace recursión sobre w. Para $w = \lambda$, se obtiene fácilmente que

$$(u \cdot v) \cdot w = (u \cdot v) \cdot \lambda = u \cdot v = u \cdot (v \cdot \lambda) = u \cdot (v \cdot w).$$

Para el paso recursivo, suponemos que $(u \cdot v) \cdot w = u \cdot (v \cdot w)$ y demostraremos que $(u \cdot v) \cdot (wa) = u \cdot (v \cdot (wa))$, para todo $a \in \Sigma$.

$$\begin{aligned} (u \cdot v) \cdot (wa) &= ((u \cdot v) \cdot w)a, && \text{(definición recursiva de la concatenación)} \\ &= (u \cdot (v \cdot w))a, && \text{(hipótesis recursiva)} \\ &= u \cdot ((v \cdot w)a), && \text{(definición recursiva de la concatenación)} \\ &= u \cdot (v \cdot (wa)), && \text{(definición recursiva de la concatenación).} \quad \square \end{aligned}$$

La propiedad asociativa permite escribir la concatenación $u \cdot v \cdot w = uvw$ sin necesidad de usar paréntesis. En general, la concatenación de n cadenas u_1, u_2, ..., u_n de Σ^* (en ese orden) se puede escribir sin ambigüedad como $u_1 u_2 \cdots u_n$.

1.4. Potencias de una cadena

Dado un alfabeto Σ y una cadena $u \in \Sigma^*$, se define u^n, con $n \in \mathbb{N}$, en la siguiente forma:

$$\begin{aligned} u^0 &= \lambda, \\ u^n &= \underbrace{uu \cdots u}_{n \text{ veces}}, \text{ si } n \geq 1. \end{aligned}$$

Lo anterior también se puede escribir como:

$$u^n = \begin{cases} \lambda, & \text{si } n = 0, \\ \underbrace{uu\cdots u}_{n \text{ veces}}, & \text{si } n \geq 1. \end{cases}$$

Es decir, u^n es la concatenación de u consigo misma n veces, si $n \geq 1$. Puesto que n es un número natural, u^n se puede definir inductivamente sobre n:

$$\begin{aligned} u^0 &= \lambda, \\ u^{n+1} &= u^n \cdot u, \text{ para todo } n \geq 0. \end{aligned}$$

Un caso particular importante se presenta cuando la cadena u consta de un solo símbolo, por ejemplo, $u = a$, donde $a \in \Sigma$. Se tiene entonces

$$\begin{aligned} a^0 &= \lambda, \\ a^n &= \underbrace{aa\cdots a}_{n \text{ veces}}, \text{ si } n \geq 1. \end{aligned}$$

Ejemplo Sea $\Sigma = \{a, b, c, d\}$. Usando la propiedad asociativa y la potenciación de cadenas de longitud 1, podemos escribir, por ejemplo:

$$\begin{aligned} &baaaddb = ba^3d^2b. \\ &dddbbcccccaa = d^3b^2c^5a^2. \\ &ababbbc = abab^3c = (ab)^2b^2c. \end{aligned}$$

1.5. Reflexión de una cadena

Dado Σ, la *reflexión* (o *inversa* o *reverso*) de una cadena $u \in \Sigma^*$ se denota u^R y se define descriptivamente así:

$$u^R = \begin{cases} \lambda, & \text{si } u = \lambda, \\ a_n \cdots a_2 a_1, & \text{si } u = a_1 a_2 \cdots a_n. \end{cases}$$

Es decir, u^R se obtiene escribiendo los símbolos de u de derecha a izquierda. De la definición se observa claramente que la reflexión de la reflexión de una cadena es la misma cadena, es decir,

$$(u^R)^R = u, \qquad \text{para } u \in \Sigma^*.$$

También es fácil verificar que $(uv)^R = v^R u^R$ para todas las cadenas $u, v \in \Sigma^*$.

> ✎ Algunos autores escriben u^{-1} en lugar de u^R para denotar la reflexión de una cadena u.

Ejercicios de la sección 1.5

① Dar una definición recursiva de u^R.

② Generalizar la propiedad $(uv)^R = v^R u^R$ a la concatenación de n cadenas, $n \geq 2$.

1.6. Subcadenas, prefijos y sufijos

Una cadena v es una *subcadena* o una *subpalabra* de u si existen cadenas x, y tales que $u = xvy$. Nótese que x o y pueden ser λ y, por lo tanto, la cadena vacía es una subcadena de cualquier cadena y toda cadena es subcadena de sí misma (ya que $u = \lambda u \lambda$).

Un *prefijo* de u es una cadena v tal que $u = vw$ para alguna cadena $w \in \Sigma^*$. Se dice que v es un *prefijo propio* de u si $v \neq u$.

Similarmente, un *sufijo* de u es una cadena v tal que $u = wv$ para alguna cadena $w \in \Sigma^*$. Se dice que v es un *sufijo propio* de u si $v \neq u$.

Obsérvese que λ es un prefijo y un sufijo de toda cadena u ya que $u\lambda = \lambda u = u$. Por la misma razón, toda cadena u es prefijo y sufijo de sí misma.

Ejemplo Sean $\Sigma = \{a, b, c, d\}$ y $u = bcbaadb$.

Prefijos de u :	Sufijos de u :
λ	λ
b	b
bc	db
bcb	adb
$bcba$	$aadb$
$bcbaa$	$baadb$
$bcbaad$	$cbaadb$
$bcbaadb$	$bcbaadb$

Algunas subcadenas de u: $baad$, aa, cba, db.

1.7. Orden lexicográfico entre cadenas

Si los símbolos de un alfabeto Σ tienen un orden preestablecido, sobre el conjunto Σ^* de todas las cadenas se puede definir un orden total, denominado *orden lexicográfico entre cadenas*. Concretamente, sea $\Sigma = \{s_1, s_2, \ldots, s_k\}$ un alfabeto dado en el cual los símbolos tienen un orden predeterminado, $s_1 < s_2 < \cdots < s_k$. En el conjunto Σ^* de todas las cadenas se define el orden lexicográfico, también denotado $<$, de la siguiente manera. Sean u, v dos cadenas en Σ^*,

$$u = a_1 a_2 \cdots a_m, \quad \text{donde } a_i \in \Sigma, \text{ para } 1 \leq i \leq m.$$
$$v = b_1 b_2 \cdots b_n, \quad \text{donde } b_i \in \Sigma, \text{ para } 1 \leq i \leq n.$$

Se define $u < v$ si

(1) $|u| < |v|$ (es decir, si $m < n$) o,

(2) $|u| = |v|$ (es decir, si $m = n$) y para algún índice i, $1 \leq i \leq m$, se cumple que

$$a_1 = b_1, a_2 = b_2, \ldots, a_{i-1} = b_{i-1}, a_i < b_i.$$

Es otros términos, las cadenas de Σ^* se ordenan inicialmente por longitud y si las cadenas u y v tienen la misma longitud, $|u| = |v| = n$, se establece el orden

$$a_1a_2\cdots a_{i-1}a_ia_{i+1}\cdots a_n < a_1a_2\cdots a_{i-1}b_ib_{i+1}\cdots b_n,$$

cuando $a_i < b_i$. Se define también $\lambda < u$, para toda cadena no vacía u.

El orden lexicográfico entre cadenas es un orden lineal, esto es, para todo par de cadenas diferentes u, v en Σ^* se tiene $u < v$ o $v < u$, y es el orden utilizado para listar las palabras en los diccionarios de los lenguajes naturales.

Ejemplo Sea $\Sigma = \{a, b, c\}$, donde los símbolos tienen el orden preestablecido $a < b < c$. Las primeras cadenas de Σ^* listadas en el orden lexicográfico son:

$\lambda, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, aba, abb, abc, aca, acb, acc, baa, bab,$
$bac, bba, bbb, bbc, bca, bcb, bcc, caa, cab, cac, cba, cbb, cbc, cca, ccb, ccc, aaaa, aaab, aaac, \ldots$

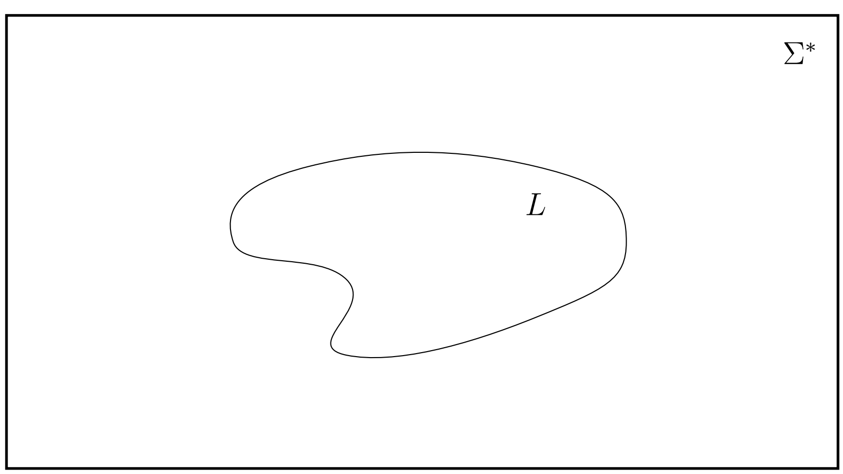
Ejercicios de la sección 1.7

Sea $\Sigma = \{a, b, c, d\}$, donde los símbolos tienen el orden preestablecido $a < b < c < d$. Para cada cadena dada u en Σ^* hallar la cadena que sigue inmediatamente a u en el orden lexicográfico.

(1) $u = dbd$.
(2) $u = acbd$.
(3) $u = dabcdd$.
(4) $u = dcaddd$.

1.8. Lenguajes

Un *lenguaje* L sobre un alfabeto Σ es un subconjunto de Σ^*, es decir $L \subseteq \Sigma^*$.



En otras palabras, un lenguaje es un conjunto de cadenas. Hay dos casos extremos:

$$\begin{array}{ll} L = \varnothing, & \text{lenguaje vacío.} \\ L = \Sigma^*, & \text{lenguaje de todas las cadenas sobre } \Sigma. \end{array}$$

Todo lenguaje L satisface $\varnothing \subseteq L \subseteq \Sigma^*$, y puede ser finito o infinito. Los lenguajes se denotan corrientemente con letras mayúsculas $A, B, C, \ldots, L, M, N, \ldots$.

Si P es una propiedad referente a las cadenas de Σ^*, el lenguaje L de todas las cadenas que satisfacen la propiedad P se denotará como $L = \{u \in \Sigma^* : P(u)\}$.

Ejemplos Los siguientes son ejemplos de lenguajes sobre los alfabetos especificados.

- $\Sigma = \{a, b, c\}$. $L = \{a, aba, aca\}$.
- $\Sigma = \{a, b, c\}$. $L = \{\lambda, a, aa, aaa, \ldots\} = \{a^n : n \geq 0\}$.
- $\Sigma = \{a, b, c\}$. $L = \{ac, abc, ab^2c, ab^3c, \ldots\} = \{ab^nc : n \geq 0\}$.
- $\Sigma = \{a, b\}$. $L =$ lenguaje de todas las cadenas de longitud $2 = \{u \in \Sigma^* : |u| = 2\} = \{aa, ab, ba, bb\}$.
- $\Sigma = \{a, b, c\}$. $L =$ lenguaje de todas las cadenas de longitud $2 = \{u \in \Sigma^* : |u| = 2\}$. Este lenguaje tiene nueve cadenas; explícitamente, $L = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$.
- $\Sigma = \{a, b, c, \ldots, x, y, z, \acute{a}, \acute{e}, \acute{\imath}, \acute{o}, \acute{u}, \tilde{n}, A, B, C, \ldots, \ldots, X, Y, Z, \acute{A}, \acute{E}, \acute{I}, \acute{O}, \acute{U}, \tilde{N}\}$. Sea L el lenguaje de todas las palabras listadas en el diccionario RAE (Real Academia Española). L es un lenguaje finito.
- $\Sigma = \{a, b, c\}$. $L = \{u \in \Sigma^* : u \text{ no contiene el símbolo } c\}$. Por ejemplo, $abbaab \in L$ pero $abbcaa \notin L$.
- $\Sigma = \{0, 1\}$. $L =$ conjunto de todas las secuencias binarias que contienen un número impar de unos.
- $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. El conjunto $\mathbb{N}$ de los números naturales se puede definir como un lenguaje sobre Σ, en la siguiente forma:

$$\mathbb{N} = \{u \in \Sigma^* : u \neq \lambda \text{ y } (u = 0 \text{ ó } 0 \text{ no es un prefijo de } u)\}.$$

Como ejercicio, el estudiante puede definir el conjunto de los enteros

$$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$$

como un lenguaje sobre un alfabeto adecuado.

1.9. Operaciones entre lenguajes

Puesto que los lenguajes sobre Σ son subconjuntos de Σ^*, las operaciones usuales entre conjuntos son también operaciones válidas entre lenguajes. Así, si A y B son lenguajes sobre Σ (es decir $A, B \subseteq \Sigma^*$), entonces los siguientes también son lenguajes sobre Σ:

$$\begin{array}{ll} A \cup B = \{u \in \Sigma^* : u \in A \vee u \in B\} & \text{Unión.} \\ A \cap B = \{u \in \Sigma^* : u \in A \wedge u \in B\} & \text{Intersección.} \\ A - B = \{u \in \Sigma^* : u \in A \wedge u \notin B\} & \text{Diferencia.} \\ \overline{A} = \Sigma^* - A = \{u \in \Sigma^* : u \notin A\} & \text{Complemento.} \end{array}$$

La diferencia $A - B$ también se suele escribir como $A \setminus B$. En estas definiciones se utilizan las conectivas lógicas $\vee$ (disyunción no-excluyente) y $\wedge$ (conjunción). Otra conectiva útil es $\veebar$ (disyunción excluyente) con la cual se define la llamada diferencia simétrica:

$$\begin{aligned} A \triangledown B &= \{u \in \Sigma^* : u \in A \veebar u \in B\} = \{u \in \Sigma^* : u \in A \text{ o } u \in B, \text{pero no ambas}\} \\ &= \{u \in \Sigma^* : \text{ o bien } u \in A, \text{ o bien } u \in B, \text{pero no ambas}\} \\ &= (A \cup B) - (A \cap B) = (A - B) \cup (B - A). \end{aligned}$$

Estas operaciones entre lenguajes se llaman *operaciones conjuntistas* o *booleanas* para distinguirlas de las *operaciones lingüísticas* (concatenación, potenciación, reflexión, clausura) que son extensiones a los lenguajes de las correspondientes operaciones entre cadenas.

1.10. Concatenación de lenguajes

La *concatenación* de dos lenguajes A y B sobre Σ, notada $A \cdot B$ o simplemente AB se define como

$$A \cdot B = AB = \{uv : u \in A, \ \ v \in B\}.$$

En general, $AB \neq BA$.

Ejemplo Si $\Sigma = \{a, b, c\}$, $A = \{a, ab, ac\}$, $B = \{b, b^2\}$, entonces

$$\begin{array}{l} AB = \{ab, ab^2, ab^2, ab^3, acb, acb^2\} = \{ab, ab^2, ab^3, acb, acb^2\}. \\ BA = \{ba, bab, bac, b^2a, b^2ab, b^2ac\}. \end{array}$$

Nótese que AB tiene cinco cadenas en total ya que la cadena ab^2 se obtiene de dos maneras distintas. Por otro lado, BA consta de seis cadenas en total.

Ejemplo Si $\Sigma = \{a, b, c\}$, $A = \{ba, bc\}$, $B = \{b^n : n \geq 0\}$, entonces

$$\begin{aligned} AB &= \{bab^n : n \geq 0\} \cup \{bcb^n : n \geq 0\}. \\ BA &= \{b^n ba : n \geq 0\} \cup \{b^n bc : n \geq 0\} \\ &= \{b^{n+1}a : n \geq 0\} \cup \{b^{n+1}c : n \geq 0\} \\ &= \{b^m a : m \geq 1\} \cup \{b^m c : m \geq 1\}. \end{aligned}$$

Propiedades de la concatenación de lenguajes. Sean A, B, C lenguajes sobre Σ, es decir $A, B, C \subseteq \Sigma^*$. Entonces

1. $A \cdot \varnothing = \varnothing \cdot A = \varnothing$.

2. $A \cdot \{\lambda\} = \{\lambda\} \cdot A = A$.

3. Propiedad Asociativa,
$$A \cdot (B \cdot C) = (A \cdot B) \cdot C.$$

4. Distributividad de la concatenación con respecto a la unión,
$$\begin{gathered} A \cdot (B \cup C) = A \cdot B \cup A \cdot C. \\ (B \cup C) \cdot A = B \cdot A \cup C \cdot A. \end{gathered}$$

5. Propiedad distributiva generalizada. Si $\{B_i\}_{i \in I}$ es una familia indexada de lenguajes sobre Σ (o sea, $B_i \subseteq \Sigma^*$ para todo $i \in I$), entonces
$$\begin{gathered} A \cdot \bigcup_{i \in I} B_i = \bigcup_{i \in I} (A \cdot B_i), \\ \Big(\bigcup_{i \in I} B_i\Big) \cdot A = \bigcup_{i \in I} (B_i \cdot A). \end{gathered}$$
Por ejemplo, si el conjunto de índices I es $\mathbb{N} = \{0, 1, 2, \ldots\}$, podemos escribir
$$A \cdot \bigcup_{i \in \mathbb{N}} B_i = A \cdot (B_0 \cup B_1 \cup B_2 \cup \cdots) = A \cdot B_0 \cup A \cdot B_1 \cup A \cdot B_2 \cup \cdots = \bigcup_{i \in \mathbb{N}} (A \cdot B_i).$$

Demostración.

1. $A \cdot \varnothing = \{uv : u \in A,\ v \in \varnothing\} = \varnothing$.

2. $A \cdot \{\lambda\} = \{uv : u \in A,\ v \in \{\lambda\}\} = \{u : u \in A\} = A$.

3. Se sigue de la asociatividad de la concatenación de cadenas.

4. Caso particular de la propiedad general, demostrada a continuación.

5. Demostración de la igualdad $A \cdot \bigcup\limits_{i \in I} B_i = \bigcup\limits_{i \in I} (A \cdot B_i)$:
$$\begin{aligned} x \in A \cdot \textstyle\bigcup_{i \in I} B_i &\iff x = u \cdot v, \quad \text{con } u \in A \ \& \ v \in \textstyle\bigcup_{i \in I} B_i \\ &\iff x = u \cdot v, \quad \text{con } u \in A \ \& \ v \in B_j, \text{ para algún } j \in I \\ &\iff x \in A \cdot B_j, \quad \text{para algún } j \in I \\ &\iff x \in \textstyle\bigcup_{i \in I} (A \cdot B_i). \end{aligned}$$

La igualdad $\Big(\bigcup\limits_{i \in I} B_i\Big) \cdot A = \bigcup\limits_{i \in I} (B_i \cdot A)$ se demuestra de forma similar. $\square$

✎ La propiedad asociativa permite escribir concatenaciones de tres o más lenguajes sin necesidad de usar paréntesis. Por ejemplo, ABC o $ABCD$.

✎ En general, no se cumple que $A \cdot (B \cap C) = A \cdot B \cap A \cdot C$. Es decir, la concatenación no es distributiva con respecto a la intersección. Contraejemplo: Sea $\Sigma = \{a\}$, $A = \{a, \lambda\}$, $B = \{\lambda\}$, $C = \{a\}$. Se tiene que $A \cdot (B \cap C) = \{a, \lambda\} \cdot \varnothing = \varnothing$. Por otro lado,
$A \cdot B \cap A \cdot C = \{a, \lambda\} \cdot \{\lambda\} \cap \{a, \lambda\} \cdot \{a\} = \{a, \lambda\} \cap \{a^2, a\} = \{a\}$.

Ejercicios de la sección 1.10

① Dar un ejemplo de un alfabeto Σ y dos lenguajes diferentes A, B sobre Σ tales que $AB = BA$.

② Dar un ejemplo de un alfabeto Σ y tres lenguajes A, B, C sobre Σ, diferentes entre sí, tales que $A \cdot (B \cap C) = A \cdot B \cap A \cdot C$.

③ Sea Σ un alfabeto dado. Se ha demostrado en esta sección, mediante un contraejemplo, que la igualdad $A \cdot B \cap A \cdot C = A \cdot (B \cap C)$ no es una identidad válida para todos los lenguajes $A, B, C \subseteq \Sigma^*$. Demostrar, sin embargo, que la contenencia

$$A \cdot (B \cap C) \subseteq A \cdot B \cap A \cdot C$$

es siempre válida para todos los lenguajes $A, B, C \subseteq \Sigma^*$.

1.11. Potencias de un lenguaje

Dado un lenguaje A sobre Σ, $(A \subseteq \Sigma^*)$, y un número natural $n \in \mathbb{N}$, se define A^n en la siguiente forma

$$\begin{aligned} &A^0 = \{\lambda\}, \\ &A^n = \underbrace{AA \cdots A}_{n \text{ veces}} = \{u_1 \cdots u_n : u_i \in A, \text{ para todo } i, \ 1 \leq i \leq n\}, \text{ para } n \geq 1. \end{aligned}$$

Las potencias de A se pueden definir por inducción sobre n:

$$\begin{aligned} A^0 &= \{\lambda\}, \\ A^{n+1} &= A^n \cdot A, \text{ para todo } n \geq 0. \end{aligned}$$

Se tiene entonces que $A^1 = A$ y A^2 es el conjunto de las concatenaciones dobles de cadenas de A:

$$A^2 = \{uv : u, v \in A\}.$$

A^3 está formado por las concatenaciones triples:

$$A^3 = \{uvw : u, v, w \in A\}$$

En general, A^n es el conjunto de todas las concatenaciones de n cadenas de A, de todas las formas posibles:

$$A^n = \{u_1 \cdots u_n : \quad (\forall i, 1 \leq i \leq n)(u_i \in A)\}.$$

1.12. La clausura de Kleene de un lenguaje

La *clausura de Kleene* o *estrella de Kleene* o simplemente la *estrella* de un lenguaje A, $A \subseteq \Sigma^*$, es la unión de todas las potencias de A y se denota por A^*.

(Descripción 1)

$$\boxed{A^* = \bigcup_{n\geq 0} A^n = A^0 \cup A^1 \cup A^2 \cup \cdots \cup A^n \cdots}$$

Según la definición de las potencias de una lenguaje, A^* consta de todas las concatenaciones de cadenas de A consigo mismas, de todas las formas posibles. Tenemos así una descripción explícita de A^*:

(Descripción 2)

$$\boxed{\begin{aligned} A^* &= \text{conjunto de } \textit{todas} \text{ las concatenaciones} \\ & \text{ de cadenas de } A\text{, incluyendo } \lambda \\ &= \{u_1 \cdots u_n : \quad (\forall i, 1 \leq i \leq n)(u_i \in A), \quad n \geq 0\}. \end{aligned}}$$

De manera similar se define la *clausura positiva* de un lenguaje A, $A \subseteq \Sigma^*$, denotada por A^+.

$$\boxed{A^+ = \bigcup_{n\geq 1} A^n = A^1 \cup A^2 \cup \cdots \cup A^n \cdots}$$

A^+ se puede describir explícitamente de la siguiente manera

$$\boxed{\begin{aligned} A^+ &= \text{conjunto de } \textit{todas} \text{ las concatenaciones de cadenas de } A \\ &= \{u_1 \cdots u_n : \quad (\forall i, 1 \leq i \leq n)(u_i \in A), \quad n \geq 1\}. \end{aligned}}$$

También es posible dar una definición recursiva de A^*:

(1)	$\lambda \in A^*$.
(2)	Si $u \in A^*$ y $v \in A$ entonces $u \cdot v \in A^*$.

En definitiva, A^* es todo lo que se puede obtener comenzando con λ y concatenando con cadenas de A. La definición recursiva de A^+ es similar, excepto que sus elementos básicos son las cadenas de A:

(1)	Si $u \in A$ entonces $u \in A^+$.
(2)	Si $u \in A^+$ y $v \in A$ entonces $u \cdot v \in A^+$.

Ejemplos Sea $\Sigma = \{a, b, c\}$.

$$\begin{aligned}
\{a\}^* &= \{\lambda, a, a^2, a^3, \ldots\}. \\
\{a\}^+ &= \{a, a^2, a^3, \ldots\}. \\
\{cb, ba\}^* &= \{\lambda, cb, ba, cbba, bacb, (cb)^2, (ba)^2, cbcbba, cbbacb, cbbaba, (cb)^3, \\
&\qquad bacbba, babacb, bacbcb, (ba)^3, \ldots\}. \\
\{a, b, c\}^* &= \{\lambda, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, baa, bab, bac, \ldots\} \\
&= \text{Lenguaje de todas las cadenas sobre el alfabeto } \Sigma.
\end{aligned}$$

✎ Dado un alfabeto Σ, se han presentado dos definiciones de Σ^*:

Σ^* = conjunto de las cadenas sobre el alfabeto Σ.

Σ^* = conjunto de todas las concatenaciones de cadenas de Σ, considerando a Σ como el lenguaje de las cadenas de longitud 1.

No hay conflicto de notaciones porque las dos definiciones anteriores de Σ^* dan lugar al mismo conjunto.

Propiedades de $*$ y $+$. Sea A un lenguaje sobre Σ, es decir, $A \subseteq \Sigma^*$.

1. $A^* = A^+ \cup \{\lambda\}$.
2. $A^* = A^+$ si y solamente si $\lambda \in A$.
3. $A^+ = A^* \cdot A = A \cdot A^*$.
4. $A^* \cdot A^* = A^*$.
5. $(A^*)^n = A^*$, para todo $n \geq 1$.
6. $A^+ \cdot A^+ \subseteq A^+$. En general, $A^+ \cdot A^+ \neq A^+$.
7. $(A^*)^* = A^*$.
8. $(A^*)^+ = A^*$.
9. $(A^+)^* = A^*$.
10. $(A^+)^+ = A^+$.

Demostración.

Las propiedades 1 y 2 se siguen inmediatamente de las definiciones de A^* y A^+.

3.
$$\begin{aligned} A \cdot A^* &= A \cdot (A^0 \cup A^1 \cup A^2 \cup \cdots) \\ &= A^1 \cup A^2 \cup A^3 \cup \cdots \\ &= A^+. \end{aligned}$$

Similarmente se demuestra que $A^* \cdot A = A^+$.

4. Si $x \in A^* \cdot A^*$, entonces $x = u \cdot v$, con $u \in A^*$, $v \in A^*$. De modo que, $x = u \cdot v$, con $u = u_1 u_2 \cdots u_n$, donde $u_i \in A$ para $1 \leq i \leq n$, $n \geq 0$, y $v = v_1 v_2 \cdots v_m$, donde $v_i \in A$ para para $1 \leq i \leq m$, $m \geq 0$. Se deduce que

$$x = u \cdot v = u_1 \cdot u_2 \cdots u_n \cdot v_1 \cdot v_2 \cdots v_m.$$

Por lo tanto, x es una concatenación de $n + m$ cadenas de A. Así que $x \in A^*$.

Recíprocamente, si $x \in A^*$, entonces $x = x \cdot \lambda \in A^* \cdot A^*$.

Hemos probado que $x \in A^* \cdot A^* \Longleftrightarrow x \in A^*$, lo cual establece la igualdad.

5. Se sigue de la propiedad 4.

6. La demostración de $A^+ \cdot A^+ \subseteq A^+$ es similar a la demostración de $A^* \cdot A^* \subseteq A^*$ presentada en la propiedad 4, pero tomando $m, n \geq 1$.

Para exhibir un contraejemplo de $A^+ \cdot A^+ = A^+$, basta considerar $\Sigma = \{a\}$ y $A = \{a\}$. Se tiene que

$$A^+ = A^1 \cup A^2 \cup \cdots = \{a\} \cup \{aa\} \cup \{aaa\} \cup \cdots = \{a^n : n \geq 1\}.$$
$$A^+ \cdot A^+ = \{a, a^2, a^3, \dots\} \cdot \{a, a^2, a^3, \dots\} = \{a^2, a^3, a^4, \dots\} = \{a^n : n \geq 2\}.$$

Se observa que $A^+ \cdot A^+ \neq A^+$.

7.
$$\begin{aligned} \big(A^*\big)^* &= \big(A^*\big)^0 \cup \big(A^*\big)^1 \cup \big(A^*\big)^2 \cup \cdots \\ &= \{\lambda\} \cup A^* \cup A^* \cup A^* \cup \cdots \\ &= A^*. \end{aligned}$$

8.
$$\begin{aligned} \big(A^*\big)^+ &= \big(A^*\big)^1 \cup \big(A^*\big)^2 \cup \big(A^*\big)^3 \cup \cdots \\ &= A^* \cup A^* \cup A^* \cup \cdots \\ &= A^*. \end{aligned}$$

9.
$$\begin{aligned} \big(A^+\big)^* &= \big(A^+\big)^0 \cup \big(A^+\big)^1 \cup \big(A^+\big)^2 \cup \cdots \\ &= \{\lambda\} \cup A^+ \cup A^+A^+ \cup \cdots \\ &= A^* \cup \ (\text{conjuntos contenidos en } A^+) \\ &= A^*. \end{aligned}$$

10.
$$\begin{aligned} \big(A^+\big)^+ &= \big(A^+\big)^1 \cup \big(A^+\big)^2 \cup \big(A^+\big)^3 \cup \cdots, \\ &= A^+ \cup \ (\text{conjuntos contenidos en } A^+) \\ &= A^+. \end{aligned}$$

Otras propiedades de $*$. Sean A, B lenguajes sobre Σ, es decir, $A, B \subseteq \Sigma^*$.

1. Si $A \subseteq B$ entonces $A^* \subseteq B^*$.
2. $(A \cup B)^* = (A^*B^*)^* = (B^*A^*)^*$.

<u>*Demostración*</u>.

1. Se sigue inmediatamente de las definiciones de A^* y B^*.

2. Demostraremos la igualdad $(A \cup B)^* = (A^*B^*)^*$. La contenencia $\subseteq$ se sigue de

$$\begin{aligned} A \subseteq A^*, \quad B \subseteq B^* &\Longrightarrow A \subseteq A^*B^*, \quad B \subseteq A^*B^* \\ &\Longrightarrow A \cup B \subseteq A^*B^* \\ &\Longrightarrow (A \cup B)^* \subseteq (A^*B^*)^* \quad \text{por la propiedad 1.} \end{aligned}$$

Ahora deduciremos la contenencia $\supseteq$. Puesto que $A \subseteq A \cup B$ y $B \subseteq A \cup B$, de la propiedad 1 se obtiene que $A^* \subseteq (A \cup B)^*$ y $B^* \subseteq (A \cup B)^*$; se concluye que $A^*B^* \subseteq (A \cup B)^*(A \cup B)^* = (A \cup B)^*$. Por la propiedad 1 se tiene entonces que

$$(A^*B^*)^* \subseteq \big((A \cup B)^*\big)^* = (A \cup B)^*. \quad \square$$

Ejercicios de la sección 1.12

Sean A, B lenguajes sobre Σ, es decir, $A, B \subseteq \Sigma^*$. Explicar por qué las siguientes igualdades no son válidas en general:

① $(A \cup B)^* = A^* \cup B^*$.

② $(A \cup B)^* = A^* \cup B^* \cup A^*B^* \cup B^*A^*$.

1.13. Reflexión o inverso de un lenguaje

Dado A un lenguaje sobre Σ, se define A^R de la siguiente forma:

$$A^R = \{u^R : u \in A\}.$$

A^R se denomina la *reflexión*, o el *inverso* de A, o el *reverso* de A.

Propiedades. Sean A y B lenguajes sobre Σ, es decir, $A, B \subseteq \Sigma^*$.

1. $(A \cdot B)^R = B^R \cdot A^R$.
2. $(A \cup B)^R = A^R \cup B^R$.
3. $(A \cap B)^R = A^R \cap B^R$.
4. $\left(A^R\right)^R = A$.
5. $(A^*)^R = \left(A^R\right)^*$.
6. $(A^+)^R = \left(A^R\right)^+$.

Demostración. Demostraremos las propiedades 1 y 5; las demás se dejan como ejercicio (opcional) para el estudiante.

1.
$$\begin{aligned} x \in (A \cdot B)^R &\Longleftrightarrow x = u^R, \quad \text{donde } u \in A \cdot B \\ &\Longleftrightarrow x = u^R, \quad \text{donde } u = vw,\ v \in A, w \in B \\ &\Longleftrightarrow x = (vw)^R, \quad \text{donde } v \in A, w \in B \\ &\Longleftrightarrow x = w^R v^R, \quad \text{donde } v \in A, w \in B \\ &\Longleftrightarrow x \in B^R \cdot A^R. \end{aligned}$$

5.
$$\begin{aligned} x \in (A^*)^R &\Longleftrightarrow x = u^R,\ \text{donde } u \in A^* \\ &\Longleftrightarrow x = (u_1 \cdot u_2 \cdots u_n)^R,\ \text{donde los } u_i \in A,\ n \geq 0 \\ &\Longleftrightarrow x = u_n^R \cdot u_2^R \cdots u_1^R,\ \text{donde los } u_i \in A,\ n \geq 0 \\ &\Longleftrightarrow x \in (A^R)^*. \end{aligned}$$

Ejercicios de la sección 1.13

① Demostrar las propiedades 2, 3, 4 y 6 de la reflexión de cadenas.

② ¿Se pueden generalizar las propiedades 2 y 3 anteriores para uniones e intersecciones de un número arbitrario de conjuntos, respectivamente?