

Apuntes de Lenguajes Formales

Leonardo H. Añez Vladimirovna¹

*Universidad Autónoma Gabriel René Moreno,
Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones,
Santa Cruz de la Sierra, Bolivia*

17 de mayo de 2019

¹Correo Electrónico: toborochi98@outlook.com

Agradecimiento a `marmot`

Notas del Autor

Estos apuntes fueron realizados durante mis clases en la materia INF319 (**Lenguajes Formales**), acompañados de referencias de libros, fuentes y código que use a lo largo del curso, en el período I-2019 en la Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones.

Para cualquier cambio, observación y/o sugerencia pueden enviarme un mensaje al siguiente correo:

`toborochi98@outlook.com`

Índice general

1. Preliminares Formales	3
1.1. Conjuntos	3
1.1.1. Conjunto Finito e Infinito	3
1.2. Preliminares	3
1.2.1. Alfabeto	3
1.2.2. Palabra	3
1.2.3. Notaciones	6
1.2.4. Cantidad de Ocurrencias	6
1.2.5. Concatenación	6
1.2.6. Inversa	7
1.2.7. Potencia de una Palabra	7
1.2.8. Principio de Inducción para Σ^*	7
1.2.9. Lenguajes	8
1.2.10. Expresiones Regulares	8
1.2.11. Módulos	9
1.2.12. Máquinas	10

Capítulo 1

Preliminares Formales

1.1. Conjuntos

1.1.1. Conjunto Finito e Infinito

Equivalencia

Dado A y B (conjuntos) los llamamos *equivalentes* si existe una biyección: $f : A \rightarrow B$

Conjunto Finito

Un conjunto A es finito si es equivalente a $\{1, 2, 3, \dots, n\}$ para algún $n \in \mathbb{N}$.

Conjunto Infinito

Un conjunto es infinito si no es finito. Si no es equivalente a $\{1, 2, 3, \dots, n\}$ es decir no hay biyección. Sin embargo no todos los conjuntos finitos son equivalentes.

- **Conjunto Contablemente Infinito:** Se dice que un conjunto es contablemente infinito si es equivalente con \mathbb{N} .
- **Conjunto Contable:** Es contable si es finito o contablemente infinito.
- **Conjunto Incontable:** Se dice que es incontable si no es contable.

Principio de las Casillas

Si A y B son conjuntos finitos no vacíos y $|A| > |B|$ entonces no existe una función inyectiva de: $A \rightarrow B$.

1.2. Preliminares

1.2.1. Alfabeto

Un alfabeto Σ es cualquier conjunto finito no vacío.

Ejemplo(s)

$$\begin{aligned}\Sigma_1 &= \{Leo, Martha\} \\ \Sigma_2 &= \{0, 1, 2, 3, \dots, 13\} \\ \Sigma_3 &= \{a, b\} \\ \Sigma_4 &= \{R, G, B, A\}\end{aligned}$$

1.2.2. Palabra

Una palabra sobre Σ es una sucesión finita de símbolos de Σ . Es decir:

$$(\sigma_1, \sigma_2, \dots, \sigma_n); \sigma \in \Sigma \quad \text{ó} \quad \sigma_1 \sigma_2 \sigma_3 \dots \sigma_n; \sigma \in \Sigma$$

Ejemplo(s)

Sobre Σ_1	Sobre Σ_2	Sobre Σ_3	Sobre Σ_4
$w_1 = \text{LeoLeo}$	$w_1 = 1111110$	$w_1 = \text{bababababa}$	$w_1 = \text{ABGR}$
$w_2 = \text{MarthaLeoMartha}$	$w_2 = 11235813$	$w_2 = \text{abba}$	$w_2 = \text{RRRA}$

Denotamos por Σ^* el conjunto de todas las palabras sobre Σ .

Longitud de una Palabra

Sea w una palabra sobre Σ , es decir $w = \sigma_1\sigma_2 \dots \sigma_n; \sigma \in \Sigma$. La longitud de w es n y se denota por: $|w| = n$.

Palabra vacía

Es la sucesión vacía de símbolos de Σ y se denota por: λ .

1.2.3. Notaciones

- $\Sigma^+ = \{w \in \Sigma^* / |w| > 0\}$
- $\Sigma^k = \{w \in \Sigma^* / |w| = k\}$
- $\Sigma^0 = \{w \in \Sigma^* / |w| = 0\} = \{\lambda\}$
- $\Sigma^1 = \{w \in \Sigma^* / |w| = 1\} = \Sigma$
- $\Sigma^* = \Sigma^+ \cup \{\lambda\}$
- $\Sigma^+ = \Sigma^* - \{\lambda\}$

1.2.4. Cantidad de Ocurrencias

Sea $w \in \Sigma^*$, denotamos por $|w|_\sigma$ al número de ocurrencias del símbolo σ en la palabra w .

Ejemplo(s)

$$\Sigma = \{a, b\}$$

- $\Sigma^* = \{\lambda, a, b, aa, bb, ab, ba, aaa, \dots\}$
- $\Sigma^0 = \{\lambda\}$
- $\Sigma_1 = \Sigma = \{a, b\}$
- $\Sigma^2 = \{ab, aa, ba, bb\}$

1.2.5. Concatenación

Sea $u, v \in \Sigma^*$ tal que $u = \sigma_1\sigma_2 \dots \sigma_n, v = \epsilon_1\epsilon_2 \dots \epsilon_n$. La concatenación de u y v se define por:

$$uv = \sigma_1\sigma_2 \dots \sigma_n\epsilon_1\epsilon_2 \dots \epsilon_n$$

Definición de Recurrencia

$$| \cdot | : \Sigma^* \rightarrow \mathbb{N}$$

$$\begin{cases} |\lambda| = 0 \\ |wa| = |w| + 1 \end{cases}$$

Ejemplo(s)

$$\begin{aligned} u &= abab \\ v &= bba \end{aligned}$$

$$\begin{aligned} uv &= ababbba \\ vu &= bbaabab \end{aligned}$$

Propiedades

- $uv \neq vu$
- $(uv)w = u(vw)$
- $u\lambda = \lambda u = u$
- $|uv| = |u| + |v|$
- $|uv|_a = |u|_a + |v|_a$

1.2.6. Inversa

Si $w = \sigma_1, \sigma_2, \dots, \sigma_n \in \Sigma^n$ entonces $w' = \sigma_n, \sigma_{n-1}, \dots, \sigma_1$ se llama inversa o transpuesta de w .

Definición de Recurrencia

$$\begin{aligned} ' : \Sigma^* &\rightarrow \Sigma^* \\ \begin{cases} \lambda' = \lambda \\ (wa)' = aw' \end{cases} \end{aligned}$$

1.2.7. Potencia de una Palabra

$$w^n = \underbrace{ww \dots w}_{n-\text{veces}}$$

Definición de Recurrencia

$$\begin{aligned} ' : \Sigma^* &\rightarrow \Sigma^* \\ \begin{cases} w^0 = \lambda \\ w^{n+1} = ww^n \end{cases} \end{aligned}$$

Propiedades

- $|w^n| = n|w|$
- $w^m w^n = w^{m+n}$
- $(w^n)^m = w^{mn}$
- $\lambda^n = \lambda$

1.2.8. Principio de Inducción para Σ^*

Sea L un conjunto de palabras sobre Σ con las propiedades:

- i.) $\lambda \in L$
- ii.) $w \in L \wedge a \in \Sigma \Rightarrow wa \in L$

Entonces

$L = \Sigma^*$, (es decir, todas las palabras sobre Σ están en L .)

1.2.9. Lenguajes

Un lenguaje sobre Σ es un subconjunto de Σ^*

Operaciones

Recordemos que ya conocemos otras operaciones (Unión, Intersección, Diferencia y Complemento), para esta materia tenemos las siguientes:

- Concatenación

Sea $A, B \subseteq \Sigma^*$

$$AB = \{w \in \Sigma^* / w = xy, x \in A, y \in B\}$$

- Transposición

Sea $A \subseteq \Sigma^*$

$$A' = \{w' \in \Sigma^* / w \in A\}$$

- Estrella de Kleene

Sea $A \subseteq \Sigma^*$

$$A^* = \{w \in \Sigma^* / w = w_1 w_2 \dots w_n \text{ para algún } k \in \mathbb{N} \text{ y para algunas } w_1, w_2, \dots, w_k \in A\}$$

1.2.10. Expresiones Regulares

Las expresiones regulares (ER) sobre un alfabeto (Σ) son las palabras sobre el alfabeto $\Sigma \cup \{\}, (\, \emptyset, \cup, *\}$ tal que cumple lo siguiente:

- 1.) \emptyset y cada símbolo de Σ es una ER.
- 2.) Si α y β son ER entonces $(\alpha\beta)$ es una ER.
- 3.) Si α y β son ER entonces $(\alpha \cup \beta)$ es una ER.
- 4.) Si α es una ER entonces α^* es una ER.
- 5.) Nada mas es una ER a menos que provenga de (1.) a (4.)

Ejemplo(s)

Para $\Sigma = \{a, b\}$ podemos formar:

$$\begin{aligned} &\emptyset \\ &a \\ &b \\ &(ab) \\ &(a \cup b) \\ &((ab) \cup b) \\ &(ba)^* \\ &((ba)^* \cup (a \cup b))^* \end{aligned}$$

Lenguaje Regular

Un lenguaje es regular ssi es generado por una expresión regular.

1.2.11. Módulos

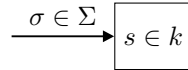
Definición

Un módulo es una tripleta $D = (k, \Sigma, f)$ donde:

- k es un conjunto finito no vacío, llamado *conjunto de estados*
- Σ es un conjunto finito no vacío, llamado *alfabeto*
- $f : k \times \Sigma \rightarrow k$, llamado *función de transición*

Interpretación

Un módulo se puede interpretar como un dispositivo que en determinados instantes de tiempo recibe señales (símbolos del alfabeto), que producen cambios en su configuración interna.



Representación

- **Tabla de Transición**
- **Grafo**

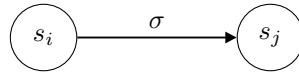
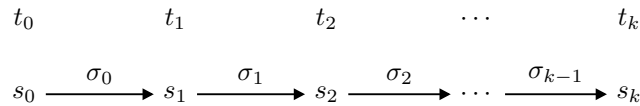


Figura 1.1: ssi: $f(s_i, \sigma) = s_j$

Comportamiento Dinámico

Sea $D = (k, \sigma, f)$ un módulo:



Función Estado Terminal

Sea $D = (k, \sigma, f)$ un módulo:

Una función de Estado Terminal del módulo D es una única función:

$$\hat{f} : k \times \Sigma \rightarrow k \text{ tal que } \forall s \in k, w \in \Sigma^*, \sigma \in \Sigma$$

$$\begin{cases} \hat{f}(s, \lambda) = s \\ \hat{f}(s, \sigma w) = \hat{f}[f(s, \sigma), w] \end{cases}$$

◊ Notas

- $w = \lambda$

$$\hat{f}(s, \sigma) = \hat{f}(s, \sigma \lambda) = \hat{f}[f(s, \sigma), \lambda] = f(s, \sigma)$$

- $\forall w \in \Sigma^*$

$$f : k \rightarrow k \text{ tal que: } f_w(s) = \hat{f}(s, w)s$$

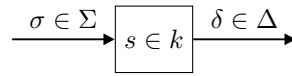
1.2.12. Máquinas

Una máquina es una quintupla $M = (k, \Sigma, \Delta, f, g)$ donde:

- k es un conjunto finito no vacío, llamado *conjunto de estados*
- Σ es un conjunto finito no vacío, llamado *alfabeto de entrada*
- Δ es un conjunto finito no vacío, llamado *alfabeto de salida*
- $f : k \times \Sigma \rightarrow k$, llamado *función de transición*
- $g : k \times \Sigma \rightarrow \Delta$, llamado *función de salida*

Interpretación

Una máquina se puede interpretar como un dispositivo que en determinados instantes de tiempo recibe señales (símbolos de entrada) que producen cambios en su configuración interna y emiten señales (símbolos de salida).



Representación

- **Tabla de Transición**
- **Grafo**

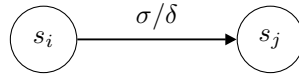


Figura 1.2: ssi: $f(s_i, \sigma) = s_j \wedge g(s_i, \sigma) = \delta$