Alfabetos, Cadenas y Lenguajes

Juan Mendivelso

Universidad Nacional de Colombia
Facultad de Ciencias
Departamento de Matemáticas

Presentación basada en las Notas de Clase del Profesor Rodrigo De Castro

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Alfabeto

Alfabeto

Conjunto finito no vacío cuyos elementos se llaman **símbolos**.

- *Notación de Símbolos:* $a, b, c, d, e, \ldots$
- *Notación de Alfabetos:* Σ, Γ, Π, $\ldots$

Ejercicio

¿Qué alfabetos conoces?

Cadena

Cadena o Palabra (String)

Una **cadena** sobre un alfabeto Σ es cualquier sucesión (secuencia) finita de elementos de Σ.

- *Notación:* $u, v, w, x, y, z, \ldots$

Cadena Vacía

La **cadena vacía** es la única cadena que no tiene símbolos y está dada sobre cualquier alfabeto Σ.

- *Notación:* λ, e, ϵ, o Λ.

Cadena

Cadena No Vacía

Una cadena no vacía u definida sobre un alfabeto Σ es de la forma $u = a_1a_2a_3 \cdots a_n$, donde $a_i \in \Sigma$ para $1 \leq i \leq n$. Además,

- u es una función $u : \{1, 2, \ldots, n\} \mapsto \Sigma$.
- El valor de $u(j)$, para $1 \leq j \leq n$, es el símbolo en la j-ésima posición of u, i.e. $u(j) = a_j$.
- Notar que el símbolo $\alpha \in \Sigma$ puede aparecer en diferentes posiciones de u; a cada una de estas se les llama **ocurrencias** de α en u.

Cadena

Longitud de una Cadena

La **longitud** de una cadena $u \in \Sigma^*$, denotada como $|u|$, se define como el número de símbolos de u incluyendo los repetidos. Más precisamente, la longitud de u es el número de posiciones que esta tiene. Es decir,

$$|u| = \begin{cases} 0, & \text{si } u = \lambda \\ n, & \text{si } u = a_1 a_2 \cdots a_n. \end{cases}$$

Conjunto de todas las Cadenas

El conjunto de todas las cadenas sobre el alfabeto Σ, incluyendo la cadena vacía, se denota como Σ^*.

Concatenación de Cadenas

Definición Descriptiva

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

1. Si $v = \lambda$, $u \cdot \lambda = \lambda \cdot u = u$.
2. Si $u = a_1 a_2 \cdots a_n$ y $v = b_1 b_2 \cdots b_m$, entonces

$$u \cdot v = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m.$$

Más formalmente, $w = uv$ si y solo si:

- $|w| = |u| + |v|$,
- $w(j) = u(j)$ para $1 \leq j \leq |u|$, y
- $w(|u| + j) = v(j)$ para $1 \leq j \leq |v|$.

Concatenación de Cadenas

Propiedades de la Concatenación

- La concatenación **es asociativa**, i.e., si $u, v, w \in \Sigma^*$, entonces

$$(uv)w = u(vw)$$

- La concatenación **no es conmutativa**. Es decir, si $u, v \in \Sigma^*$, en general $uv \neq vu$.

Ejercicio

Demostrar que la concatenación es asociativa usando la definición descriptiva.

Potenciación de Cadenas

Definición Descriptiva

Dada $u \in \Sigma^*$ y $n \in \mathbb{N}$, la cadena u^n se define descriptivamente como:

$$u^n = \begin{cases} \lambda, & \text{si } n = 0 \\ uuu \cdots u \text{ (n veces)}, & \text{si } n \geq 1. \end{cases}$$

Ejercicio

Usando la definición descriptiva, mostrar que $|w^{n+m}| = |w^n| + |w^m|$.

Reflexión de Cadenas

Definición Descriptiva

La **reflexión** o **inversa** de una cadena $u \in \Sigma^*$, denotada como u^R o u^{-1}, se define descriptivamente como:

$$u^R = \begin{cases} \lambda, & \text{si } u = \lambda \\ a_n a_{n-1} \cdots a_2 a_1, & \text{si } u = a_1 a_2 \cdots a_{n-1} a_n \end{cases}$$

Propiedades

1. $(u^R)^R = u$ para $u \in \Sigma^*$.
2. $(uv)^R = v^R u^R$ para todas las cadenas $u, v \in \Sigma^*$

Ejercicio

Generalizar la Propiedad 2 a la concatenación de n cadenas, $n \geq 2$.

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Conjunto

Definición Recursiva

Un conjunto se define recursivamente por medio de 3 cláusulas:

1. **Cláusula Básica:** Se especifican los elementos más sencillos (o básicos) de S.
2. **Cláusula Recursiva:** Se especifican 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.

Conjunto de todas las Cadenas

Definición Recursiva

Dado un alfabeto Σ, el conjunto de todas las cadenas Σ^* se define como:

1. $\lambda \in \Sigma^*$.
2. Si $u \in \Sigma^*$, entonces $ua \in \Sigma^*$ para todo $a \in \Sigma$.

Concepto sobre Cadena

Definición Recursiva

Dado un alfabeto Σ, para definir un concepto C sobre todas las cadenas en Σ^* se utilizan dos cláusulas:

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

Ejercicio

Defina longitud de cadena de manera recursiva.

Recursión sobre Cadenas

Definición Recursiva

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

1. Se demuestra $P(\lambda)$.
2. Se demuestra $P(u) \Rightarrow P(ua)$ para toda $u \in \Sigma^*$ y para todo $a \in \Sigma$.

Concatenación de Cadenas

Definición Recursiva

Sea $u \in \Sigma^*$. La concatenación de cadenas se define recursivamente como:

1. $u \cdot \lambda = \lambda \cdot u = u$.
2. $u \cdot (va) = (u \cdot v)a$, donde $a \in \Sigma$ y $v \in \Sigma^*$.

Ejercicio

Usando la definición recursiva, demostrar que la concatenación es asociativa: si $u, v, w \in \Sigma^*$, entonces $(uv)w = u(vw)$.

Potenciación de Cadenas

Definición Recursiva

Dada $u \in \Sigma^*$ y $n \in \mathbb{N}$, la cadena u^n se define inductivamente como:

1. $u^0 = \lambda$.
2. $u^{n+1} = u^n u$ para $n \geq 0$.

Reflexión de Cadenas

Ejercicio

Dar una definición recursiva de u^R.

Subcadena

Subcadena o Subpalabra (Substring)

Sean $u, v \in \Sigma^*$. La cadena v es una **subcadena** o **subpalabra** de u si existen cadenas $x, y \in \Sigma^*$ tales que $u = xvy$.

Prefijo

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

Sufijo

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

Orden Lexicográfico de Cadenas

- Sea $\Sigma = \{s_1, s_2, \ldots, s_k\}$ un alfabeto en el que los símbolos tienen un orden predeterminado: $s_1 < s_2 < \cdots < s_k$.
- En el conjunto Σ^* se define el **orden lexicográfico** de las cadenas, denotado como $<$, de la siguiente manera.
- Sea $u, v \in \Sigma^*$ tales que

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

- Entonces, se define $u < v$ si
  1. $|u| < |v|$, i.e. $m < n$, or
  2. $|u| = |v|$ y para algún índice i, $1 \leq i \leq m$, se cumple que

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

  3. $\lambda < u$, $\forall u \in \Sigma^* : |u| > 0$.

Orden Lexicográfico de Cadenas

Ejercicio

Sea $\Sigma = \{a, b, c, d\}$ un alfabeto en el que los símbolos tienen un orden predeterminado: $a < b < c < d$. Hallar la cadena que sigue a u en el orden lexicográfico:

- $u = dbd$
- $u = dabcdd$
- $u = acbd$
- $u = dcaddd$
- $u = bbb$

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Lenguaje

- Un lenguaje L sobre un alfabeto Σ es un subconjunto $L \subseteq \Sigma^*$.
- Todo lenguaje L satisface $\emptyset \subseteq L \subseteq \Sigma^*$ y es finito o infinito.
- **Notación:** $A, B, C, \ldots, L, M, N, \ldots$
- Si P es una propiedad de las cadenas de Σ^*, el lenguaje L de las cadenas que satisfacen la propiedad P se denota como $L = \{u \in \Sigma^* : P(u)\}$.

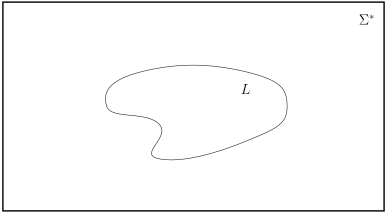

Figura 1: Lenguaje (Tomado de [1])

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Operaciones entre Lenguajes

Operaciones Conjuntistas o Booleanas

Sean $A, B \subseteq \Sigma^*$. Los siguientes también son lenguajes sobre Σ:

- **Unión:** $A \cup B = \{u \in \Sigma^* : u \in A \vee u \in B\}$.
- **Intersección:** $A \cap B = \{u \in \Sigma^* : u \in A \wedge u \in B\}$.
- **Diferencia:** $A - B = A \setminus B = \{u \in \Sigma^* : u \in A \wedge u \notin B\}$.
- **Diferencia Simétrica:** $A \triangle B = \{u \in \Sigma^* : u \in A \veebar u \in B\}$.
- **Complemento:** $\overline{A} = \Sigma^* - A = \{u \in \Sigma^* : u \notin A\}$.

Operaciones entre Lenguajes

Operaciones Lingüísticas

Las operaciones lingüísticas sobre los lenguajes son:

- Concatenación.
- Potenciación.
- Reflexión.
- Clausura.

Concatenación de Lenguajes

Definición

Sean $A, B \subseteq \Sigma^*$. La concatenación de A y B, denotada como $A \cdot B$ o AB, se define como:

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

En general, $AB \neq BA$.

Propiedades de la Concatenación

Sean $A, B, C \subseteq \Sigma^*$. Entonces,

1. $A \cdot \emptyset = \emptyset \cdot A = \emptyset$.
2. $A \cdot \{\lambda\} = \{\lambda\} \cdot A = A$.
3. **Propiedad Asociativa:** $A \cdot (B \cdot C) = (A \cdot B) \cdot C$.
4. **Distributividad con respecto a la Unión:**
$$A \cdot (B \cup C) = A \cdot B \cup A \cdot C,$$
$$(B \cup C) \cdot A = B \cdot A \cup C \cdot A.$$
5. **Distributividad Generalizada con respecto a la Unión:** Sea $\{B_i\}_{i \in \mathcal{I}}$ una familia indexada de lenguajes sobre Σ, i.e. $B_i \subseteq \Sigma^*$, entonces:
$$A \cdot \bigcup_{i \in \mathcal{I}} B_i = \bigcup_{i \in \mathcal{I}} (A \cdot B_i) \quad \text{y} \quad \left(\bigcup_{i \in \mathcal{I}} B_i \right) \cdot A = \bigcup_{i \in \mathcal{I}} (B_i \cdot A).$$

Propiedades de la Concatenación

Sean $A, B, C \subseteq \Sigma^*$. Entonces,

- La propiedad asociativa permite escribir concatenaciones sin paréntesis, i.e. ABC.
- En general, **no se cumple** que $A \cdot (B \cap C) = A \cdot B \cap A \cdot C$.
  Contraejemplo: $\Sigma = \{a\}$, $A = \{a, \lambda\}$, $B = \{\lambda\}$ y $C = \{a\}$.

Propiedades de la Concatenación

Ejercicio

1. Dar un ejemplo de un alfabeto Σ y dos lenguajes diferentes $A, B \subseteq \Sigma^*$ tales que $AB = BA$.
2. Dar un ejemplo de un alfabeto Σ y tres lenguajes A, B y C sobre Σ, diferentes entre sí, tales que $A \cdot (B \cap C) = AB \cap AC$.
3. Demuestre que $A \cdot (B \cap C) \subseteq AB \cap AC$. Muestre un ejemplo.
4. Muestre que $AB \cap AC \subseteq A \cdot (B \cap C)$ no se cumple en el caso general a través de un contraejemplo.

Potencias de un Lenguaje

Definición Descriptiva

Dado un lenguaje $A \subseteq \Sigma^*$ y un número $n \in \mathbb{N}$, se define A^n de la siguiente forma:

1. $A^0 = \{\lambda\}$.
2. $A^n = AA \cdots A(\text{n veces}) = \{\mathrm{u}_1 \cdots u_n : (\forall i)(u_i \in A), 1 \leq i \leq n\}$.

Definición Recursiva

Dado un lenguaje $A \subseteq \Sigma^*$ y un número $n \in \mathbb{N}$, se define A^n de la siguiente forma:

1. $A^0 = \{\lambda\}$.
2. $A^{n+1} = A^n \cdot A$, para todo $n \geq 0$.

Clausura de Kleene de un Lenguaje

Definición Descriptiva

La **clausura de Kleene**, **estrella de Kleene** o **estrella** de un lenguaje $A \subseteq \Sigma^*$, denotada como A^* es la unión de todas las potencias de A, i.e.

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

$$= \{u_1 \cdots u_n : (\forall i)(u_i \in A), 1 \leq i \leq n, n \geq 0\}.$$

Definición Recursiva

La **estrella** de un lenguaje $A \subseteq \Sigma^*$, denotada como A^*, se define recursivamente como:

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

Clausura Positiva de un Lenguaje

Definición Descriptiva

La **clausura positiva** de un lenguaje $A \subseteq \Sigma^*$, denotado como A^+ es:

$$\begin{aligned} A^+ &= \bigcup_{n \geq 1} A^n = A^1 \cup A^2 \cup \cdots \\ &= \{u_1 \cdots u_n : (\forall i)(u_i \in A), 1 \leq i \leq n, n \geq 1\}. \end{aligned}$$

Definición Recursiva

La **clausura positiva** de un lenguaje $A \subseteq \Sigma^*$, denotada como A^+, se define recursivamente como:

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

Propiedades de las Clausuras

Sea $A \subseteq \Sigma^*$. Entonces,

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. $(A^+)^+ = A^+$.

Propiedades de las Clausuras

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

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

En general, la igualdad $(A \cup B)^* = A^* \cup B^*$ **no es válida**.

Ejercicio

1. Demostrar las propiedades 1 y 2.
2. Mostrar ejemplos de las propiedades 1 y 2.
3. Mostrar un ejemplo en que se cumpla $(A \cup B)^* = A^* \cup B^*$.
4. Mostrar un contrajeemplo de $(A \cup B)^* = A^* \cup B^*$.

Reflexión o Inverso de un Lenguaje

Sea $A \subseteq \Sigma^*$. Se define como la **reflexión**, **reverso** o **inverso** de A, denotado como A^R, así:

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

Propiedades de la Reflexión

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. $(A^R)^R = A$.
5. $(A^*)^R = (A^R)^*$.
6. $(A^+)^R = (A^R)^+$.

Ejercicio

1. Demostrar estas propiedades.
2. ¿Se pueden extender las Propiedades 2 y 3 a uniones/intersecciones de un número arbitrario de lenguajes?

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Lenguajes Regulares

Definición Recursiva

La colección de todos los **lenguajes regulares** sobre un alfabeto Σ se define como:

1. $\emptyset$, $\{\lambda\}$ y $\{a\}$, $a \in \Sigma$, son **lenguajes regulares básicos** sobre Σ.
2. Si A y B son lenguajes regulares sobre Σ, también lo son:
    - $A \cup B$.
    - $A \cdot B$.
    - A^*.

Lenguajes Regulares

Operaciones Regulares

- Unión.
- Concatenación.
- Estrella de Kleene.

Propiedades

- Todo lenguaje finito es regular ya que se puede obtener a partir de una unión finita.
- Si $\{A_i\}_{i\in\mathcal{I}}$ es una familia infinita de lenguajes regulares, la unión de estos, i.e. $\bigcup_{i\in\mathcal{I}} A_i$, no necesariamente es regular.

Lenguajes Regulares

Ejercicio

Sea $\Sigma = \{a, b\}$. Muestre que los siguientes lenguajes son regulares:

1. Lenguaje de todas las cadenas en Σ^*.
2. Lenguaje de todas las cadenas en Σ^+.
3. Lenguaje de todas las cadenas que tienen exactamente una a.
4. Lenguaje de todas las cadenas que comienzan con una b.
5. Lenguaje de todas las cadenas que contienen la subcadena ba.
6. Lenguaje de las cadenas sobre Σ que tienen exactamente longitud 2.
7. Lenguaje de las cadenas sobre Σ que tienen al menos longitud 2.
8. Lenguaje de las cadenas sobre Σ que tienen máximo longitud 2.

Expresiones Regulares

Definición Recursiva

Las **expresiones regulares** sobre un alfabeto Σ se definen como:

1. **Expresiones Regulares Básicas:**
   - $\emptyset$ es una expresión regular que representa al lenguaje $\emptyset$.
   - λ es una expresión regular que representa al lenguaje $\{\lambda\}$.
   - a es una expresión regular que representa al lenguaje $\{a\}$, para cada $a \in \Sigma$.
2. Si R y S son expresiones regulares sobre Σ que representan a los lenguajes regulares A y B, respectivamente, entonces:
   - $(R \cup S)$ es una expresión regular que representa al lenguaje $A \cup B$.
   - (RS) es una expresión regular que representa al lenguaje $A \cdot B$.
   - $(R)^*$ es una expresión regular que representa al lenguaje A^*.

Expresiones Regulares

Otra Notación

Denótese con $L[R]$ al lenguaje representado por la expresión regular R. Las expresiones regulares se pueden presentar como:

1. **Expresiones Regulares Básicas:** $\emptyset$, λ y a (donde $a \in \Sigma$) son expresiones regulares tales que:
   - $L[\emptyset] = \emptyset$.
   - $L[\lambda] = \{\lambda\}$.
   - $L[a] = \{a\}$, para cada $a \in \Sigma$.
2. Si R y S son expresiones regulares entonces:
   - $L[(R \cup S)] = L[R] \cup L[S]$.
   - $L[(RS)] = L[R] \cdot L[S]$.
   - $L[(R)^*] = (L[R])^*$.

Expresiones Regulares

Ejercicio

Sea $\Sigma = \{a, b\}$. Construya expresiones regulares para los siguientes lenguajes:

1. Lenguaje de todas las cadenas en Σ^*.
2. Lenguaje de todas las cadenas en Σ^+.
3. Lenguaje de todas las cadenas que tienen exactamente una a.
4. Lenguaje de todas las cadenas que comienzan con una b.
5. Lenguaje de todas las cadenas que contienen la subcadena ba.
6. Lenguaje de las cadenas sobre Σ que tienen exactamente longitud 2.
7. Lenguaje de las cadenas sobre Σ que tienen al menos longitud 2.
8. Lenguaje de las cadenas sobre Σ que tienen máximo longitud 2.

Expresiones Regulares

Algunas Propiedades

- La representación de lenguajes regulares por medio de expresiones regulares no es única.
- En expresiones regulares también se permite el uso de potencias.
- Por la propiedad $A^+ = A^* \cdot A = A \cdot A^*$, se permite el uso de la clausura positiva en expresiones regulares.

Ejercicio

Encuentre otras expresiones regulares que representen el mismo lenguaje que las siguientes:

1. $(a \cup b)^*$
2. $(a^+ \cup b)ab^+$
3. $(a \cup b)^+ \cup a^+b^+$

Expresiones Regulares

Ejercicio

Sea $\Sigma = \{a, b\}$. Construya expresiones regulares para los siguientes lenguajes de las cadenas u sobre Σ tales que:

1. $u \in \Sigma^*$.
2. tienen exactamente una a.
3. comienzan con a.
4. terminan con a.
5. contienen la subcadena ba.
6. tienen longitud par.
7. tienen longitud impar.
8. su longitud es múltiplo de 3.
9. su longitud cumple que $|u| \mod 3 = 2$.

Expresiones Regulares

Algunas Propiedades

- En las expresiones regulares se usan reglas de precedencia. El orden es:
    1. $*$ Estrella de Kleene.
    2. $\cdot$ Concatenación.
    3. $\cup$ Unión.
- Sean L, R y S expresiones regulares. La propiedad $L[R(S \cup T)] = L[RS \cup RT]$ permite distribuir o "factorizar" para obtener nuevas expresiones regulares.

Expresiones Regulares

Ejercicio

Sea $\Sigma = \{a, b, c\}$. Construya expresiones regulares para los siguientes lenguajes:

1. su penúltimo símbolo es una a.
2. su número de $a$es es exactamente 2.
3. su número de $a$es es al menos 2.
4. su número de $a$es es máximo 2.
5. su número de $a$es es par.
6. no contienen dos $a$es consecutivas.
7. no contienen la subcadena bc.
8. empiezan y terminan con el mismo símbolo.
9. empiezan y terminan con diferentes símbolos.

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Expresiones Regulares en UNIX

grep

- **G**lobally search for a **R**egular **E**xpression and **P**rint matching lines.
- Funciona de forma similar a `egrep`.

egrep

- **E**xtended **G**lobal search for a **R**egular **E**xpression and **P**rint matching lines.
- Usualmente usado en consolas de Unix.
- Dado un archivo de texto `archivo.txt`, y una expresión regular `re`, ejecutar el comando: `egrep ‘‘re’’ archivo.txt`.
- La ejecución de este comando muestra las líneas que satisfacen la expresión regular señalando las partes exactas que la satisfacen.

Expresiones Regulares en UNIX

Concatenación

Escribir caracteres o palabras concatenadas uno a continuación del otro.

Unión

.	cualquier caracter
\w	caracteres que forman palabras
\W	caracteres que no forman palabras
[a+d]	cualquier caracter en $a \cup + \cup d$
[a-c]	cualquier caracter en $a \cup b \cup c$
[ˆabd]	cualquier caracter que no pertenezca a $(a \cup b \cup d)$
[ˆa-c]	cualquier caracter que no pertenezca a $(a \cup b \cup c)$
$a \vert b$	cualquier caracter en $(a \cup b)$
$Homero \vert Marge$	que sea Homero o Marge

Expresiones Regulares en UNIX

Más de Unión

Los conjuntos de caracteres de POSIX agrupan grupos de caracteres:

[:alpha:]	caracteres alfabéticos
[:lower:]	caracteres alfabéticos en minúscula
[:upper:]	caracteres alfabéticos en mayúscula
[:digit:]	dígitos
[:xdigit:]	dígitos incluyendo hexadecimales
[:alnum:]	caracteres alfanuméricos
[:punct:]	signos de puntuación
[:space:]	espacios en blanco
[:blank:]	espacios en blanco, tabulaciones, etc
[:print:]	caracteres imprimibles

Expresiones Regulares en UNIX

Repetición

p?	cero o una ocurrencia de *p*
p+	una o más ocurrencias consecutivas de *p*
p*	cero o más ocurrencias consecutivas de *p*
(pa)*	cero o más ocurrencias consecutivas de *pa*
p{2}	dos *p*es consecutivas
p{2,4}	de 2 a 4 *p*es consecutivas
p{2,}	mínimo dos *p*es consecutivas
p{,4}	máximo dos *p*es consecutivas
(pp){,4}	máximo 4 ocurrencias consecutivas de *pp*

Expresiones Regulares en UNIX

Caracteres de escape

- Algunos caracteres tienen una función especial en la búsqueda de expresiones regulares. Por ejemplo: [] () {} . * ?.
- Para referirse a estos como tal, se deben preceder de \

Palabras y Líneas

^a	líneas que comiencen por *a*
a$	líneas que terminen en *a*
\< *a*	palabras que comienzan por *a*
a \>	palabras que terminan en *a*

Outline

1. Alfabetos y Cadenas
   - Definiciones Básicas de Alfabetos y Cadenas
   - Definiciones y Demostraciones Recursivas
2. Lenguajes
   - Definición
   - Operaciones entre Lenguajes
3. Lenguajes Regulares
   - Lenguajes y Expresiones Regulares
   - Expresiones Regulares en UNIX
   - Aplicaciones de de Expresiones Regulares

Análisis Léxico

Analizador Léxico

- Un analizador léxico realiza las siguientes operaciones:
  1. Examina el código fuente.
  2. Reconoce los tokens: subcadenas de caracteres que constituyen una unidad lógica; por ejemplo, palabras reservadas e identificadores.
- Las expresiones regulares son útiles para describir los tokens.
- Antes de estas herramientas basadas en expresiones regulares, la implementación de analizadores léxicos tomaba meses; ahora días.
- La modificación del analizador léxico usualmente consiste en la modificación de alguna de las expresiones regulares.

Análisis Léxico

Los comandos `lex` y `flex` de Unix/GNU convierten expresiones regulares en autómatas finitos deterministas para generar una función eficiente que parte el código fuente en tokens.

```
else                    {return(ELSE);}

[A-Za-z][A-Za-z0-9]*    {code to enter the found identifier
                         in the symbol table;
                         return(ID);
                        }

>=                      {return(GE);}

=                       {return(EQ);}

...
```

Figura 2: Fragmento de entrada para `lex`. (Tomado de [3])

Análisis Léxico

- Las expresiones regulares también son útiles para el análisis léxico de números de punto flotante, cadenas de caracters, comentario y palabras reservadas.
- Se construye un autómata que acepte la unión de estas expresiones regulares, de manera que se pueda:
  - saber si algún elemento se reconoce.
  - determinar qué se reconoció.
  - priorizar a qué categoría se asigna a un elemento. Por ejemplo, `else`.

Búsqueda de Patrones en Texto

- Las expresiones regulares útiles para definir patrones de interés.
- No se han desarrollado herramientas tan potentes como las del análisis léxico.
- Se empieza por una definición general del patrón a buscar. Luego se refina con casos más específicos.
- Se puede usar para
  - reconocer páginas Web.
  - crear listas de correo.
  - clasificar negocios por su locación.
  - reconocer direcciones.

```
'[0-9]+[A-Z]? [A-Z][a-z]*( [A-Z][a-z]*)*
(Street|St\.|Avenue|Ave\.|Road|Rd\.)'
```

Bibliografía

1. Rodrigo De Castro Korgi. **Notas de Clase de Introducción a la Teoría de la Computación**. 2023.
2. Harry R. Lewis, Christos H. Papadimitriou. **Elements of the Theory of Computation**. Prentice Hall. 1998.
3. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman. **Introduction to Automata Theory, Languages and Computation, Third Edition**. Pearson. 2006.
4. http://manpages.ubuntu.com/manpages/focal/es/man1/grep.1.html