Contents

[Preliminar. 2](#_Toc207472869)

[Alfabeto 2](#_Toc207472870)

[Palabra. 2](#_Toc207472871)

[Palabra Vacia. 2](#_Toc207472872)

[Longitud de la Palabra. 2](#_Toc207472873)

[Concatenacion. 2](#_Toc207472874)

[Principio de Induccion para ∑\* 3](#_Toc207472875)

[Definicion de Recurrencia 3](#_Toc207472876)

[INVERSA 4](#_Toc207472877)

[Definion de recurrencia 4](#_Toc207472878)

[Potencia de una palabra 5](#_Toc207472879)

[Por recurrencia 5](#_Toc207472880)

[Propiedades 5](#_Toc207472881)

[PREFIJOS SUFIJOS Y SUBPALABRAS 5](#_Toc207472882)

[LENGUAJES 5](#_Toc207472883)

[OPERACIONES ENTRE LENGUAJES 6](#_Toc207472884)

# Preliminar.

Alfabeto. Alfabeto ∑

Un alfabeto sigma es un *Conjunto finito NO Vacio* y sus elementos serán llamados **letras**.

Ejemplo: ∑1 = {Luis, Maria, Jaime} //hay 3 letras

∑2 = {a, b} //hay 2 letras

∑3 = {+, -, \*} //hay 3 letras

## Palabra.

Sea ∑ un alfabeto. Una palabra sobre ∑ es una sucesión finita de símbolos de ∑.

Es decir w = S1, S2, …, Sn ; Si ∈ ∑ // w = palabra

Ejemplo: ∑ = {0, 1}

Solucion: w1 = 01 //longitud = 2 w4 = 111101 //longitud = 6

W2 = 10 //longitud = 2 w5 = 0 //longitud = 1

W3 = 00000 //longitud = 5 w6 = 1 //longitud = 1

Ejemplo: ∑ = {0, 10}

Solucion: w1 = 0 10 //longitud = 2 w2 = 10 10 0 //longitud = 3

## Palabra Vacia.

Sea ∑ un alfabeto. La palabra vacia es la sucesión vacia de símbolos de ∑ y se denota por: λ

## Longitud de la Palabra.

Sea ∑ un alfabeto y sea w ∈ σ1, σ2, …, σn; σi ∈ ∑, se dice es la longitud de la palabra w y se denota por: |w| = n

Ejemplo: ∑ = {a, b} w = bbbba

Solución: |w| = 5 | λ | = 0

Ejemplo: ∑ = {Joquin, Saturnino, Fabiola} w = Joquin Saturnino Fabiola

Solución: |w| = 3

NOTACIONES.

Vamos a denotar como ∑ \*. Es el conjunto de todas las palabras sobre ∑, excepto la palabra vacia.

∑\* El conjunto de todas las palabras sobre ∑ de longitud K.

|  |  |  |
| --- | --- | --- |
| ∑\* | ∑\* = ∑+U {λ} |  |
| ∑+ | ∑+ = ∑\* - {λ} |
| ∑K | ∑K = {w ∈ ∑\* / |w| = K} |

∑2 = {w ∈ ∑\* / |w| = 2}

∑2 = {w ∈ ∑\* / |w| = 1}

∑0 = {w ∈ ∑\* / |w| = 0}

∑0 = {λ}

∑3 = {w ∈ ∑\* / |w| = 3}

Ejemplo: ∑ = {a, b}

Solucion: ∑\* = {λ, a, b, aa, bb, ab, ba, aaa, bbb, …}

∑2 = {aa, bb, ab, ba} //22 = 4

∑1 = {a, b} //21 = 2

∑3 = {aaa, bbb, abb, aab, aba, baa, bab, bba} //23 = 8

## Concatenacion.

Sea ∑ un alfabeto y sea u = σ1, σ2, …, σn y v = ζ1, ζ2,.., ζn; σi,ζi ∈ ∑

Se define la concatenación de u y v por: uv = σ1, σ2, …, σn ζ1, ζ2,.., ζn; σi,ζi

Ejemplo: ∑ = {a, b} u = bba v = ab

Solución: uv = bba ab son distiintos uv <> vu |u| = 3 |v|=2

vu = ab bba |uv| = 5

* uv ≠ vu
* (uv)w = u(vw)
* uλ = u = λu
* |uv| = |u|+||v|

Denotamos por: Al numero de ocurrencias de a el w.

|w|σ = al numero de ocurrencias de σ en la palabra w

Ejemplo: ∑ = {a, b} u = bba

Solucion: |u|a = 1 |u|b = 2

Ejercicio: ∑ = {a, b}

1. A3 = {w ∈ ∑\* / |w|=3 ^ |w|a = 2}
2. A4 = {w ∈ ∑\* / |w|=4 ^ |w|a = 2}
3. A5 = {w ∈ ∑\* / |w|=5 ^ |w|a = 2}
4. An = {w ∈ ∑\* / |w|=n ^ |w|a = 2}

Se pide:

1. Escribir a cada conjunto A3, A4, A5, An por extensión
2. |Ai| = ? para i=3,4,5,n

Solucion:

a)

1. A3​ = {aaa, bbb, **aab**, **aba**, abb, **baa**, bba, bab, }.

A3 = { aab, aba, baa} //3 palabras

1. A4 = {aaaa, bbbb, aaab, abba, abbb, abab, baaa, bbaa, baba, … }.

A4 = {aabb, abab, abba, baab, baba, bbaa}. //6 palabras

1. A5 = {aaaaa, bbbbb, …}

A5 = {aabbb, ababb, abbab, abbba, baabb, babab, babba, bbaab, bbaba, bbbaa}. //10 palabras

1. An = {biabj−i−1abn−j : 1≤i<j≤n}

b)

∣A3​∣=3,  ∣A4​∣=6,  ∣A5​∣=10,  ∣An​∣=

## Principio de Induccion para ∑\*

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

1. λ ∈ L
2. λ ∈ L ^ a ∈ ∑ ⇒ wa ∈ L

Entonces L =∑\* (es decir todas las palabras de ∑\* están en L).

1. 1 ∈ P
2. K ∈ P ⇒ k +1 ∈ P

La definición de longitud no nos sirve para la demostración.

### Definicion de Recurrencia

| | : ∑\* → N // los naturales comienzan desde el 0, ya que |λ|=0

1. **|λ|=0**
2. **|wa|=|w|+1**
3. **|uwa| = |u|+|wa|**

Ejemplo: ∑ = {a, b} w = bbabb

Solución: |w| = 5

Demostrando: |w| = |bbabb| def. de recurrencia

= |bbab|+1 def. de recurrencia

= |bba|+1+1 def. de recurrencia

= |bb|+1+1+1 def. de recurrencia

= |b|+1+1+1+1 operando

= |b|+4 def. de concatenacion

= | λb |+4 def. de recurrencia

= | λ |+1+4 def. de |λ|=0

= 0+1+4 operando

= 5

Demostrar: |uv|=|u|+|v|; ∀u,v ∈ ∑\*

∑ = {a, b}; u = abba; v = baab

Solucion: |uv| = |uv| =

iguales

= |abba baab| = | abba |+| baab |

= 8 = 4 + 4

= 8

Por tanto, para demostrar correctamente usamos el principio de inducción

**PRUEBA**: |uv| = |u|+|v|; ∀u,v ∈ ∑\*

* Primero definimos L:
* Tomamos la variable v para mas facilidad:

L = {v ∈∑\* / |uv|=|u|+|v|}

/\* comentario

Sea V = {x / x es vocal}

x es vocal ⇒ x ∈ V

si z ∈ V ⇒ z es vocal

\*/

1. ¿ λ ∈ L?

**|u λ |** = |u| = |u|+0 = **|u|+| λ |**

|u λ | = |u|+| λ |

∴ λ ∈ L

Por demostrar: w ∈ L ^ a ∈ ∑ ⇒ wa ∈ L

Sea w ∈ L ^ a ∈ ∑

|uw| = |u|+|w| ^ a ∈ ∑ (H.I) Hipotesis Inductiva

Por demostrar: wa ∈ L

Por demostrar: |u(wa)|=|u|+|wa|

|u(wa)| Asociativa

= |(uw)a| HI

= |(uw)|+1 Def de concatenacion

= (|u|+|w|) +1 Asociativa, ya que u,w son números

= |u|+(|w|+1) Def de recurrencia

= |u(wa)| Def de concatenacion

= |u|+|wa|

∴ wa ∈ L

Por demostración Condicional: w ∈ L ^ a ∈ ∑ ⇒ wa ∈ L

ENTONCES: L = ∑\*

## INVERSA

Sea u = a1, a2, …,an ∈ ∑\* a la palabra u prima (u’)

u’ = an, an-1, .., a2, a1 se llama inversa o transpuesta de u, (es decir a la escrita de orden inverso).

Ejemplo: ∑ = {a, b} u=bba

Solución: u’ = abb

Definion de recurrencia. ‘: ∑\* →∑\*

1. **λ’ =λ**
2. **(wa)’=aw’**

Ejemplo: u =bba u’=?

Solucion: u’ = (bba)’ Def de inversa

= abb’ Def de concatenacion

= ab(λb)’ Def de inversa

= abbλ’ Def de recurrencia **λ’ =λ**

**=** abb

Ejercicio: Demostrar |w’| = |w|; ∀w ∈ ∑\*

Solucion:

Llamamos L = {w ∈ ∑\* / |w’|=w}

1. Por definición (λ)’ = λ luego | λ’|=| λ| ⇒ λ ∈ L
2. Sea u ∈ L ^ a ∈ ∑

Por definición (ua)’ = au’ , luego

|(ua)’| def de inversa

=|au’| def de concatenacion

=|a|+|u’|

Aplicamos hipótesis de inducción, w ∈ L, tenemos |u’| = |u|

Entonces: |(ua)’| def de inversa

= |au’| def de concatenación

= |a|+|u’| H.I. u’=u

= |a|+|u| conmutativa

= |u|+|a|

Sucesivamente por propiedad de longitud |ua|=|u|+|a|

Luego |(ua)’|

= |u|+|a|

= |ua| osea ua ∈ L

∴ L ∈ ∑\*

**Si queremos verificar:**

**|w’| = |w|; ∀w ∈ ∑\***

**w=hola**

|w’|= |hola|

=|hol|+1 def recureencia

=|ho|+1+1 def recurencia

=|h|+1+1+1 def recurencia

=| λh|+4 def h= λh

=| λ|+1+4 def recurencia

=0+5 def | λ|=0

=5

Ejercicio: Demostrar: |uv| = |u’|+|v’|; ∀w ∈ ∑\*

L={v ∈ ∑\*/|uv| = |u’|+|v’|}

#### Crecimiento por izquierda

(u)’ ∈ L ^ a ∈ ∑ ⇒ aw ∈ ∑

## Potencia de una palabra

wn = w\*w\*w…w

w1 = w

w0 = λ

### Por recurrencia

w0 = λ

wn+1 = w\*wn

Ejemplo: ∑={a, b} w=ba

**Solucion:**

w4 = ba ba ba ba

w4 = w w3

= ba w w2

= ba ba w2

= ba ba w w

= ba ba ba ba

# Propiedades

1. **|wn| = n|w|**
2. **wnwm = wn+m**
3. **(wn)m = wn+m**
4. **λn=λ**

Ejercicios: Demostraciones

Demostrar las propiedades desde a) hasta d) y buscar palabras u,v sobre ∑={a, b} tales que u2v2 ≠ (uv)2

# PREFIJOS SUFIJOS Y SUBPALABRAS

Sean v,z ∈ ∑\*

1. Se dice que v es prefijo de z si y solo si existe w ∈∑\* tal que z=vw y se escribe v pref z
2. Se dice que v es sufijo de z si y solo si existe u ∈∑\* tal que z=uv y se escribe v suf z
3. Se dice que v es subpalabra de z si y solo si existe u1u2 ∈∑\* tal que z=u1vu2 y se escribe v subp z

Ejemplo: ∑ = {a, b}

**Solucion:**

z = babbab

v = bab w = bab

v = b w = abb ab

v = λ w = babbab

v = babbab w= λ

Ejercicio

1. Cuantos prefijo, sufijo, subpalabras, tiene la palabra z=a1,a2,…,an ∈ ∑
2. Demostrar:

X pref y ^ y pref x ⇒ x = y

X pref y ^ y pref z ⇒ x pref z

**Solucion:**

Demostracion Condicional: Sobre el antecendente, mostra el concecuente

# LENGUAJES

**Definicion matematica**. Sea ∑ un alfabeto. Un lenguaje sobre ∑ es un subconjunto de ∑\*

Ejemplo:

∅ //El vacio es subconjunto de ∑\*

∑\* //El ∑\* es subconjunto de ∑\*

Ejemplo: ∑ = {a, b} ∑\* = {λ, a, b, aa, ba, ab, bb, …}

Solucion:

L1 = {ab} L4 = ∑+

L2 = { λ} L5 = { λ, a, aa, aaa, …}

L3 = { λ, a, aa, b} L5 = {w ∈ ∑\* / w = an, n ∈ N}

Ejemplo: ∑ = {a, b} L = { w ∈ ∑\* / |w| = 5, |w|a = 2}

Solucion:

OPERACIONES ENTRE LENGUAJES. Sean A,B, ∈ ∑\*

**Union**. A U B = {w∈∑\* / w∈A v w∈B}

**Interseccion**. A ∩ B = {w∈∑\* / w∈A ^ w∈B}

**Diferencia**. A - B = {w∈∑\* / w∈A ^ w∉B}

**Complemento**. AC = {w∈∑\* / w∉A}

-------------------------------------------------------------------------------------------------------------------------------------------------------------

**Concatenacion**. AB = {w∈∑\* /w=xy, x∈A ^ y∈B}

**Transposicion**. A’ = {w’∈∑\* / w∈A}

**Estrella de kleene**. A\* = {w∈∑\* / w = w1, w2,…,wk, para algunas w11, w21,…,wk ∈ A para algún k ∈ N}

**Ejemplo**:

1. Sea ∑={a, b} y sea **P={a, ab}; P={ λ, a, ba}**
2. PUQ e) P’ i) PC
3. P∩Q f) Q’ j) QC
4. P-Q g) PQ
5. Q-P h) P2=PP

**Solucion:**

1. {λ, a, ab, ba} e) {a, ba} h) {aa, aab, aba, abab}
2. {a} f) {λ, a, ab} i) PC = ∑\*- P = ∑\* - {a, ab}
3. {ab} g) {aλ, aa, ba, abλ, aba, abba} = j) QC = ∑\* - Q = ∑\* - {λ, a, ba}
4. {λ, ba} {a, aa, ba, ab, aba, abba}

1. Sea L1, L2, L3 ⊆ ∑\*. Demostrar:
2. L1 Փ = Փ L1 = A
3. L1 {λ} = {λ} L1 = L1
4. L1 (L2L3) = (L1L2) L3
5. (L1UL2) L3 = L1L3U L2L3
6. L1 (L2UL3) = L1L2U L1L3
7. Sean P,Q,R ⊆ ∑\*. Demostrar
8. (PUQ)’ = P’UQ’
9. (PQ)’ = Q’P’
10. (PUQ)2 = P2UPQUQPUQ2
11. Dar ejemplo de lenguaje P,Q,R sobre ∑ = {a,b} tales que

(P∩Q)R ≠ (PR) ∩ (QR)

¿Cuál de las inclusiones es valida?

**Ejemplo**: Sean A, B ⊆ ∑\*. Demostrar

A ⊆ AUB

**Solucion:**

PRUEBA.

* Por demostrar: w∈A ⇒ w∈(AUB)

Sea w∈A Regla Premisa

= w∈A v w∈B Adicion

= w∈AUB Union

= w∈A ⇒ w∈(AUB) Demostracion Condicional

= A⊆B Inclusion

**Ejemplo**: Sean A, B ⊆ ∑\*. Demostrar

(A-B)C = ACUB

Solucion:

PRUEBA

* Por demostrar: (A-B)C = ACUB
* Por demostrar: (A-B)C ⊆ ACUB ^ ACUB ⊆ (A-B)C

1. II)

* Por demostrar: **I)** w∈ (A-B)C ⇒ w∈ACUB

1. w ∈(A-B)C Regla premisa
2. w ∉ (A-B)C  Complemento
3. ~[w∈ (A-B)] Notacion
4. ~[w∈A ^ w∉B] Diferencia
5. w∉A v w∈B Morgan
6. w∈AC v w∈B Complemento
7. w∈ACUB Union
8. w ∈(A-B)C ⇔ w ∈ ACUB Demostracion Condicional **1al 7**
9. (A-B)C ⊆ ACUB Incucion

* Por demostrar: **II)** w∈ACUB ⇒ w∈ (A-B)C

1. w ∈ ACUB Regla Premisa
2. w∈AC v w∈B Union
3. w∉A v w∈B Complemento
4. ~ (w∈A ^ w∉B) Morgan
5. ~ [w∈(A-B)] Diferencia
6. w∈(A-B)C Complemento
7. w∈ACUB ⇔ w∈ (A-B)C Demostracion Condicional **1 al 6**
8. ACUB ⊆ (A-B)C

