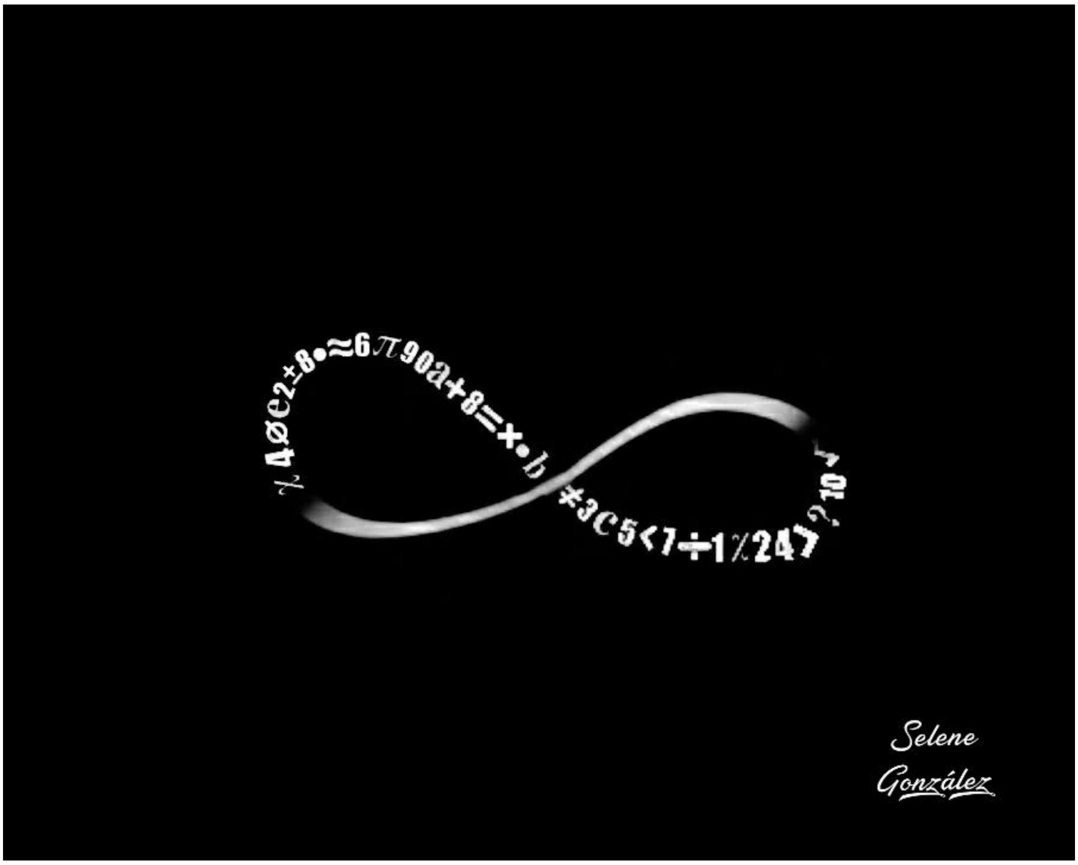
DIRECCIÓN GENERAL DE EDUCACIÓN TÉCNICO PROFESIONAL

MÉTODOS DISCRETOS



Tercera Edición

“La matemática discreta es parte fundamental para el estudio de las ciencias de la computación, puesto que las estructuras que analiza son utilizadas para modelar y resolver problemas reales mediante el diseño y programación tanto de algoritmos como de estructuras de datos”

Dr. Alberth Alvarado, director del Departamento de Matemática Aplicada en la Universidad Galileo.

A los estudiantes...

En este curso veremos distintos temas que le permitirá a Ud. entender nuevas estructuras y así generarse una idea suficientemente amplia de las mismas permitiendo modelar los problemas más importantes de las ciencias de la computación. Desde la conocida teoría de conjuntos hasta las funciones recursivas y sus aplicaciones; así como las notables soluciones que aporta la teoría de grafos. Analizaremos entre otras cosas algunos de los problemas que genera el “infinito” y como tratarlos. Se tendrá especial cuidado en el manejo de la simbología en la que está escrita nuestra teoría, es muy difícil escribir y entender sobre algo si no tenemos conocimiento pleno de los símbolos (¡y su significado!) en lo que está escrito nuestro objeto de estudio, las matemáticas discretas.

INDICE

1 Conjuntos 7
1.1 Definición de Conjunto 7
1.2 Notación de conjuntos 7
1.3 Cardinal de un conjunto 8
1.4 Subconjunto 8
1.5 Igualdad de conjuntos 8
1.6 Conjunto vacío 9
1.7 Conjunto de partes o conjunto potencia 9
1.8 Algunos conjuntos para recordar 9
1.9 Operaciones de conjuntos las leyes de las teorías de conjuntos 10
1.10 Conjuntos disjuntos 10
1.11 Conjunto complemento 10
1.12 Complemento relativo 10
1.13 Propiedades de la teoría de conjuntos 11
1.14 Técnicas de conteo y diagramas de Venn 12
1.15 Práctico de Conjuntos 13
2 Relaciones 16
2.1 Producto cartesiano 16
2.2 Relación 16
2.3 Algunas propiedades de las relaciones 18
2.4 Relación de equivalencia 19
2.5 Clases de equivalencia 20
2.6 Representación de relaciones: Matrices y Grafos dirigidos 21
2.7 Matriz 21
2.7.1 Representación de relaciones usando matrices 22
2.7.2 Representación de relaciones usando grafos dirigidos 25
2.8 Grafo dirigido 25
2.9 Práctico de Relaciones 26
3 Funciones 32
3.1 Definiciones 32
3.2 Dominio, Codominio y Conjunto Imagen 33
3.3 Función Inyectiva 33
3.4 Función Sobreyectiva 34
3.5 Función Biyectiva 34
3.6 Gráfico de una función 35
3.7 Función Identidad 36
3.8 Igualdad de Funciones 36
3.9 Función Compuesta 37
3.10 Función Inversa 37
3.11 Conjunto PreImagen 38
3.12 Práctico de Funciones 38
4 Grafos 42
4.1 Introducción 42
4.2 Terminología Básica 42
4.3 Multigrafos y Grafos Pesados 43
4.4 Definiciones 44
4.5 Resumen de las definiciones 44
4.6 Conexidad 44

4.7 Grado de un vértice.. 45
4.8 Circuitos y Recorridos Eulerianos.. 45
4.9 Práctico de Grafos 45
5 Lenguajes **49**
5.1 Esquema de inducción.. 49
5.2 Definición Inductiva de un Conjunto 50
5.3 Significado de una definición inductiva 50
5.4 Lenguajes.. 51
5.5 Pertenencia a un conjunto Inductivo 51
5.6 Práctico de Lenguajes.. 54
6 Recursividad **56**
6.1 Definiciones recursivas.. 56
6.2 Funciones definidas de forma recurrente.. 57
6.3 Definición de f Recursiva 58
6.4 Práctico Funciones Recursivas 59
7 Haskell **61**
7.1 Cuarenta Años de Diseño y Construcción.. 61
7.2 Escriba su primer programa en Haskell.. 62
7.3 Haskell como calculadora.................................... **¡Error! Marcador no definido.**
8 Bibliografía.. **82**

Símbolos y su significado.

$\in$ pertenece

U conjunto Universo

Z^+ conjunto de los enteros positivos

$\geq$ mayor o igual

$\leq$ menor o igual

$<$ menor estricto

$>$ mayor estricto

$\neq$ distinto de

$\forall$ para todo

$\exists$ existe

$\nexists$ no existe

$\Leftrightarrow$ sí y solo sí

$\subseteq$ subconjunto

\# cardinal de un conjunto

$\wedge$ conjunción

$\vee$ disyunción

$\dot{2}$ múltiplo de dos o par

$\dot{2}+1$ impar

/ tal que

$\bar{A}$ conjunto A complemento

∞ infinito

$\emptyset$ conjunto vacío

α palabra variable

ε palabra vacía

$\cup$ unión de conjuntos

$\cap$ intersección de conjuntos

$\oplus$ diferencia simétrica (conjuntos)

$\triangle$ diferencia simétrica

$\rightarrow$ Condicional (entonces)

“Si he conseguido ver más lejos, es porque me he aupado en hombros de gigantes. No se lo que pareceré a los ojos del mundo, pero a los míos es como si hubiese sido un muchacho que juega en la orilla del mar y se divierte de tanto en tanto encontrando un guijarro más pulido o un caparazón más hermoso, mientras el inmenso océano de la verdad se extendía, inexplorado frente a mí”.

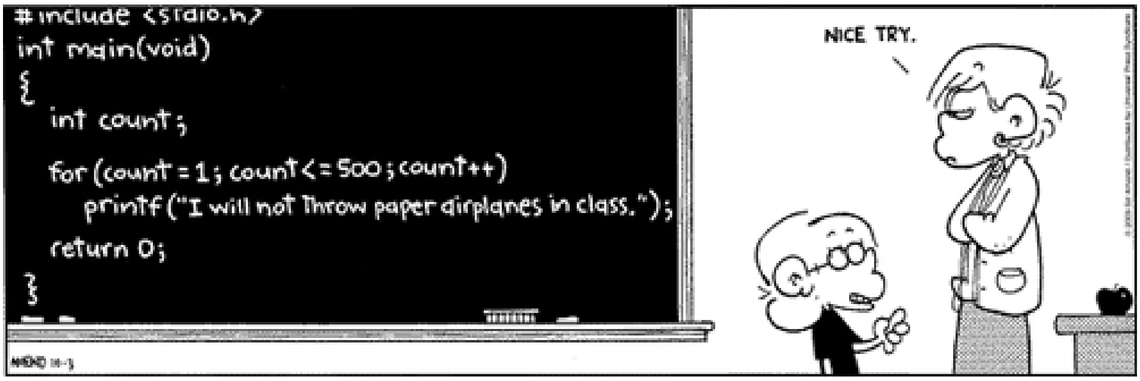
Isaac Newton

Revisado por el Sr. Inspector del Área 925 *Lic. Gonzalo Pastor*.

Agradecimientos: Agradecemos la colaboración de los siguientes profesores quienes brindaron una opinión crítica y constructiva durante la realización del presente material de estudio: Natali Paggiola, Alejandro Chmiel, José Malagrino.

Un agradecimiento especial a la Cra. María Estela Estévez Fernández por la edición del presente trabajo.

Imagen de portada: Selene González.



1 Conjuntos

1.1 Definición de Conjunto

D1. Un conjunto es una colección bien definida de objetos. Estos objetos se llaman elementos y se dice que son miembros del conjunto. Matemáticas discreta y combinatoria Capítulo 3: Teoría de conjuntos Ralph P. Grimaldi	**D2.** Un conjunto es una colección desordenada de objetos. Los objetos de un conjunto se llaman también elementos del conjunto. Se dice que un conjunto contiene a sus elementos Matemática discreta y sus aplicaciones Capítulo 1: Los fundamentos: Lógica y demostración, conjuntos y funciones Kenneth H. Rosen

Utilizaremos letras mayúsculas, A,B,C,….., para representar los conjuntos y letras minúsculas, a,b,c,d,…, para representar los elementos. Para un conjunto A, escribiremos $x \in A$ si x es un elemento de A; $y \notin A$ indica que y no es elemento de A.

1.2 Notación de conjuntos

Un conjunto puede designarse enumerando sus elementos dentro de llaves

$$A=\{1,2,3,4,5\}$$

Complete:

- El conjunto A está formado por ____
  ____
- 2 ___ A y 6 ___ A

Otra notación común para este conjunto es

$$A=\{x \;/\; x \text{ es un entero y } 1 \leq x \leq 5\}$$

Se lee "A es el conjunto de todos los x tales que x es un entero entre 1 y 5 inclusive"

Al indicar que x es un número entero, estamos especificando un **universo**, que por lo general se denota como U, sólo elegiremos elementos de U para formar nuestro conjunto.

Ejemplo.
Para el universo $U = \{1,2,3,4,5,\ldots\ldots\}$, conjunto de los enteros positivos, cuya notación es Z^+.
El conjunto $A=\{1,4,9,16,\ldots,64,81\} = \{x^2 \;/\; x \in U, x^2<100\} = \{x^2 \;/\; x \in U \wedge x^2<100\} = \{x \in U \;/\; x^2<100\}$

Veamos: para el universo U definido en el ejemplo, ¿podría escribir los conjuntos con otras notaciones?

B = {1,4,9,16} = ____

C = {2,4,3,8,10………} = ____

Los conjuntos A y B son conjuntos finitos, mientras que C es un conjunto infinito. Al trabajar con conjuntos como A y C, podemos describir los conjuntos en términos de las propiedades que deben satisfacer sus elementos, o enumerar elementos suficientes para indicar un patrón evidente.

1.3 Cardinal de un conjunto

Para cualquier conjunto finito A, **|A|** ó **#A** denota el número de sus elementos, y se conoce como **cardinal o tamaño de A**.
En el ejemplo anterior tenemos que |A| = _____ y |B| = _____

1.4 Subconjunto

Si C, D son conjuntos de un universo U, decimos que C es un ***subconjunto*** de D y escribimos $\mathbf{C \subseteq D}$, o $\mathbf{D \supseteq C}$, si cada elemento de C es un elemento de D. Si, además, D contiene un elemento que no está en C, entonces C es un ***subconjunto propio*** de D y se denota $\mathbf{C \subset D}$, o $\mathbf{D \supset C}$.

Para cualesquiera conjuntos C, D del universo U,

$$C \subseteq D \Leftrightarrow \forall x[x \in C \Rightarrow x \in D]$$

Aquí el cuantificador universal $\forall x$ indica que debemos considerar cada elemento x del universo dado U pertenecientes al conjunto C.

Para conjuntos C, D finitos ¿Qué podemos decir del |C| y del |D|?

$C \subseteq D \Rightarrow |C|$ ____ $|D|$

$C \subset D \Rightarrow |C|$ ____ $|D|$

Definimos nuestro universo U, tal que $U = \{1,2,3,4,5\}$ y consideramos el conjunto $A=\{1,2\}$ y $B=\{x / x^2 \in U\}$

¿Qué podemos decir de A y B? ____

¿$A \subseteq B$? ____

¿$B \subseteq A$? ____

1.5 Igualdad de conjuntos

Para un universo dado U, los conjuntos C y D (tomados de U) son iguales, C = D, cuando $C \subseteq D$ y $D \subseteq C$.

¿Cuándo podemos decir que $A \not\subseteq B$?

¿Cuándo podemos decir que $A \neq B$?

Teorema
Sean A $\subseteq U$, B $\subseteq U$, C $\subseteq U$

1. A $\subseteq$ B y B $\subseteq$ C $\Rightarrow$ A $\subseteq$ C
2. A $\subset$ B y B $\subseteq$ C $\Rightarrow$ A $\subset$ C
3. A $\subseteq$ B y B $\subset$ C $\Rightarrow$ A $\subset$ C
4. A $\subset$ B y B $\subset$ C $\Rightarrow$ A $\subset$ C

1.6 Conjunto vacío

El **conjunto vacío**, es el conjunto que **no contiene elementos**, se denota como $\varnothing$ o **{}**

Teorema
Para cualquier universo U, sea A $\subseteq U$, entonces $\varnothing \subseteq$ A, y si A $\neq \varnothing$, entonces $\varnothing \subset$ A

Para el conjunto C = {1,2,3} determine todos los subconjuntos de C

1.7 Conjunto de partes o conjunto potencia

Si A es un conjunto del universo U, el **conjunto de partes de A**, que se denota **$\mathcal{P}$(A),** es **el conjunto de todos los subconjuntos de A**

1.8 Algunos conjuntos para recordar

- Z = el conjunto de los enteros
- N = el conjunto de los números naturales
- Z^+ = el conjunto de los enteros positivos
- Z^- = el conjunto de los enteros negativos
- R = el conjunto de los números reales
- R^* = el conjunto de los números reales distintos de cero
- Q = el conjunto de los números racionales
- Q^* = el conjunto de los números racionales distintos de cero
- C = el conjunto de los números complejos.

Después de aprender a contar, nos enfrentamos a métodos para combinar los números contados. El primer método es la suma. La adición de dos elementos de Z^+ produce un tercer elemento de Z^+, llamado suma. Por lo tanto, nos podemos concentrar en la adición sin tener que ir más allá de Z^+. Esto también es cierto para la multiplicación.
Las operaciones de adición y multiplicación son operaciones binarias cerradas en Z^+. Cuando calculamos $a + b$, para a, b $\in Z^+$, hay dos operandos, a y b por eso la operación se llama binaria. Como $a + b \in Z^+$ si a, b $\in Z^+$, decimos que la adición en Z^+ es cerrada. Sin embargo la operación binaria de la división (con divisor distinto de cero) no es cerrada en Z^+, ya que por ejemplo, 1/2 (= 1 $\div$ 2) $\notin Z^+$, aunque 1, 2 $\in Z^+$.

Ahora presentamos algunas **operaciones binarias entre conjuntos**

1.9 Operaciones de conjuntos las leyes de las teorías de conjuntos

Para A, B $\subseteq U$ definimos lo siguiente:

A$\cup$B (la unión de A y B) = $\{x \mid x \in A \vee x \in B\}$
A$\cap$B (la intersección de A y B) = $\{x \mid x \in A \wedge x \in B\}$
A$\triangle$B ó A$\oplus$B (la diferencia simétrica de A y B) = $\{x \mid x \in A\cup B \ \wedge \ x \notin A\cap B\}$

Si $U = \{1,2,3,4,\ldots\ldots, 9,10\}$, A =$\{1,2,3,4,5\}$, B = $\{3,4,5,6,7\}$ y C = $\{7,8,9\}$
¿Qué elementos tienen los siguientes conjuntos?

A$\cap$B =
A$\cup$B =
B$\cap$C =
A$\cap$C =
A$\triangle$B =
A$\cup$C =
A$\oplus$C =

1.10 Conjuntos disjuntos

Sean S $\subseteq U$, T $\subseteq U$. Los conjuntos **S y T son disjuntos si S$\cap$T = $\varnothing$**

Teorema
Sean S $\subseteq U$, T $\subseteq U$.
S y T son disjuntos $\Leftrightarrow$ S$\cup$T = S$\triangle$T

1.11 Conjunto complemento

Para A $\subseteq U$, el complemento de A que se denota $\bar{A}$, está dado por $\{x \mid x \in U \wedge x \notin A\}$

Para los conjuntos $U = \{1,2,3,4,\ldots\ldots, 9,10\}$, A =$\{1,2,3,4,5\}$, B = $\{3,4,5,6,7\}$ y C = $\{7,8,9\}$. ¿Podría indicar el complemento de los conjuntos A, B y C?

1.12 Complemento relativo

Para A, B $\subseteq U$, el complemento relativo de A en B, que se denota B – A esta dado por $\{x \mid x \in B \wedge x \notin A\}$

Para los conjuntos $U = \{1,2,3,4,\ldots\ldots, 9,10\}$, A =$\{1,2,3,4,5\}$, B = $\{3,4,5,6,7\}$ y
C = $\{7,8,9\}$. Determine:

B – A =	A – B =	A – C =
C – A =	A – A =	U – A =

Teorema

Para cualquier universo U y cualesquiera conjuntos A, E $\subseteq U$ las siguientes proposiciones son equivalentes:

a) A $\subseteq$ E   b) A$\cup$E = E

c) A$\cap$E = A   d) $\bar{E} \subseteq \bar{A}$

1.13 Propiedades de la teoría de conjuntos

Identidad	*Nombre*
$A \cup \emptyset = A$ $A \cap U = A$	Leyes de identidad
$A \cup U = U$ $A \cap \emptyset = \emptyset$	Leyes de dominación
$A \cup A = A$ $A \cap A = A$	Leyes idempotentes
$\overline{(\bar{A})} = A$	Ley de complementación
$A \cup B = B \cup A$ $A \cap B = B \cap A$	Leyes conmutativas
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Leyes asociativas
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Leyes distributivas
$\overline{A \cup B} = \bar{A} \cap \bar{B}$ $\overline{A \cap B} = \bar{A} \cup \bar{B}$	Leyes de De Morgan
$A \cup (A \cap B) = A$ $A \cap (A \cup B) = A$	Leyes de absorción
$A \cup \bar{A} = U$ $A \cap \bar{A} = \emptyset$	Leyes de complemento

Cuando analizamos las relaciones que pueden existir entre los conjuntos implicados en una proposición de igualdad o contenido entre conjuntos, podemos estudiar la situación de manera gráfica.

Un diagrama de Venn (llamado así en honor del lógico inglés John Venn, 1834 – 1923) se construye de la manera siguiente: U aparece como el interior de un rectángulo, mientras

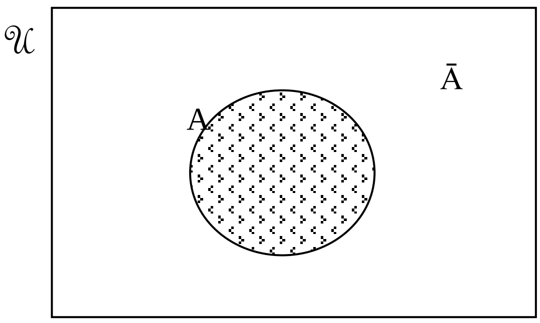


Figura 1.1

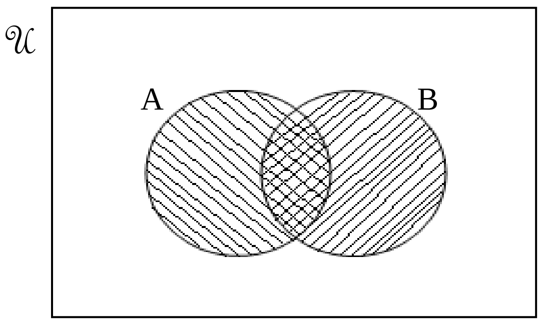


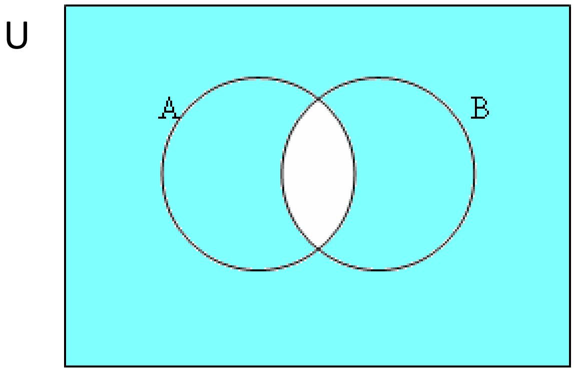
Figura 1.2

que los subconjuntos de *U* se representan mediante los interiores de círculos y otras curvas cerradas.

La zona sombreada de la Figura 1.1 representa el conjunto A y el área no sombreada representa el complemento de A.

En la Figura 1.2 la región sombreada representa A∪B. El conjunto A∩B es el área cuadriculada de la figura.

¿Podría indicar que representa en cada caso el área sombreada?

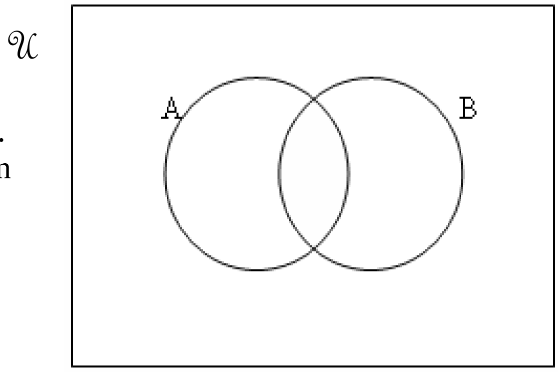


1.14 Técnicas de conteo y diagramas de Venn

Para los conjuntos A y B, de un universo finito, los diagramas de Venn nos ayudan a obtener fórmulas de conteo para $|\bar{A}|$ y |A∪B| en términos de |A| , |B| y |A∩B|

Ejemplo

En una clase de 50 alumnos de primer año, 30 estudian C++, 25 estudian Pascal y 10 están estudiando ambos lenguajes. ¿Cuántos alumnos de primer año estudian algún lenguaje de computación?



1.15 Práctico de Conjuntos

1. a) ¿Cuáles de los siguientes conjuntos son iguales?

A={1,2,3} B={3,2,1,3} C={3,1,2,3}

b) Para U $= \{1,2,3,\ldots,9,10\}$, $A = \{x/x = \dot{2} \ \wedge \ x \geq 6\}$, B = {1,2,3,4}, C = {1,2,3,5,7} con $A, B, C \subseteq$ U contestar lo que sigue:

a) $\emptyset \neq \{\emptyset\}$ V F b) $2 \in B$ V F c) $2 \notin C$ V F

d) #A= 3 V F e) $B \subseteq A$ V F f) $C = \left\{x / x = \dot{2} + 1 \wedge 1 < x \leq 7\right\}$ V F

g) $\emptyset \subset$ B V F h) $A = \{6, 8, 10\}$ V F

2. Determine todos los elementos de cada uno de los siguientes conjuntos

a. A={1+(-1)^n / n∈N}

b. B={n+(1/n) / n∈{1,2,3,5,7}}

c. C={n^3+n^2 / n∈{0,1,2,3,4}}

d. D={1/(n^2+n) / n∈Z$^+$, n es impar, n≤11}

3. ¿Cuáles de los siguientes conjuntos son no vacíos?

a. A={x/x∈N, 2x+7=3}

b. B={x/x∈Q, x^2+4=6}

c. C={x/x∈R, x^2+5=4}

d. D={x/x∈Z, 3x+5=9}

e. E={x/x∈R, x^2+4=6}

f. F={x/x∈R, x^2+3x+3=0}

4. Consideremos los subconjuntos de Z

A = {2m + 1 / m ∈ Z} C = {2p – 3 / p ∈ Z } E = {3s + 2 / s ∈ Z}

B = {2n + 3 / n ∈ Z} D = {3r + 1 / r ∈ Z} C = {3t – 2 / t ∈ Z }

¿Cuáles de las siguientes proposiciones son verdaderas y cuáles falsas?

a) A = B c) B = C e) D = F

b) A = C d) D = E f) E = F

5. Para U = {1,2,3,…, 9,10}, A ={1,2,3,4,5}, B = {1,2,4,8}, C = {1,2,3,5,7} y D = {2,4,6,8}. Determine:

a) (A∪B)∩C d) C∩D g) (B – C) – D

b) A∪(B∩C) e) (A∪B) – C h) B – (C – D)

c) C∪D f) A∪(B – C) i) (A∪B) – (C∩D)

6. Determine los conjuntos

a) A y B cuando A – B = {1,3,7,11}, B – A = {2,6,8} y A∩B = {4,9}

b) C y D cuando C – D ={1,2,4}, D – C ={7,8} y C∪D = {1,2,4,5,7,8,9}

7. Sea un universo U finito con A, B $\subseteq U$. Ordene la siguiente lista en orden creciente de acuerdo con el tamaño.

a) $|A\cup B|$, $|B|$, $|\varnothing|$, $|A\cap B|$, $|U|$

b) $|A - B|$, $|\varnothing|$, $|A|$, $|A| + |B|$, $|A\cup B|$, $|U|$

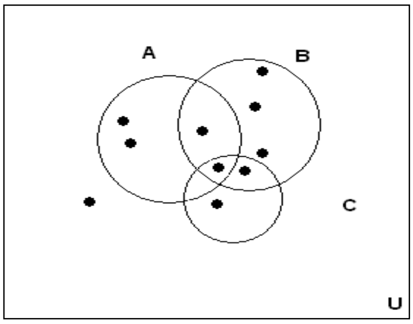
c) $|A - B|$, $|A| + |B|$, $|\varnothing|$, $|A \triangle B|$, $|A\cup B|$, $|U|$

8. Dado un universo U y los conjuntos A, B $\subseteq U$
¿a qué es igual $(A - B) \cap (B - A)$? Demuestre su afirmación.

9. En una reunión de científicos hay un total de 120 científicos, hay un conjunto de 100 matemáticos y 35 físicos. ¿Cuántos científicos son matemáticos y físicos a la vez? Plantee el problema en el lenguaje de conjuntos.

10. De 100 estudiantes, 32 estudian matemática, 20 estudian física, 45 estudian biología, 15 estudian matemática y biología, 7 estudian matemática y física, 10 estudian física y biología, 30 no estudian ninguna de estas tres materias.
Encuentre el número de estudiantes que estudian las tres materias
Encuentre el número de estudiantes que estudian exactamente una de las tres Materias.

11. Dado el siguiente diagrama, determine:



$\#((A\cup B)\cap C =$

$\#((A-B)) =$

$\#(A^C) =$

$\#(A\cap B\cap C) =$

$\#((A\cup C)-B) =$

$\#(A\oplus B) =$

12. Simplifique las siguientes expresiones:

i)

$$\left\{\left\{\left[(P\cap Q)\cup R\right]\cup\overline{\left[(R\cup P)\cap(R\cup Q)\right]}\right\}\cap\left\{\left[(P\cup Q)\cap P\right]\cup\left[(P\cup Q)\cap Q\right]\right\}\right\}\cap P$$

ii) $\left\{\overline{\left[(A\cup B)\cap A\right]}\cup A\right\}\cap\left\{\left[(A\cup C)\cap C\right]\cup\overline{\left(\overline{A}\right)}\right\}$

13. Un total de 180 ómnibus de una compañía de transporte de pasajeros desean cubrir tres recorridos con el siguiente detalle:

a) El recorrido uno (R1) cuenta con 82 unidades, el dos (R2) con 113 unidades y el tres (R3) con 70 unidades.

b) Quince unidades cubren (solo) R1 y R2, 18 (solo) R2 y R3.

c) Treinta unidades cubren las tres líneas.

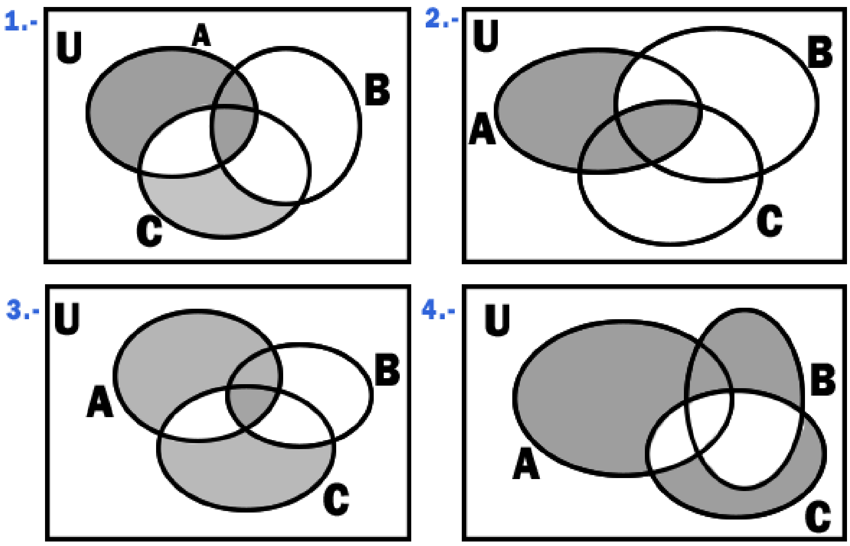
d) Diez no cubren ninguna de las líneas.

i. ¿Cuántos ómnibus cubren solamente los recorridos uno y tres?

ii. ¿Cuántas unidades cubren en forma exclusiva cada uno de los recorridos?

14. Una población de 500 hormigas toma dos caminos distintos para salir del hormiguero; 200 toman el camino 1 (solamente) y 150 los dos caminos (para salir y entrar). ¿Cuántas hormigas usan solamente el camino 2?

15. Representar mediante lenguaje de conjuntos los siguientes diagramas:



2 Relaciones

2.1 Producto cartesiano

Para los conjuntos A, B $\subseteq U$, el **producto cartesiano** de A y B, se denota **A×B** siendo **A×B= {(a,b) / a ∈ A, b ∈ B}**

Decimos que los elementos de A×B son pares ordenados. Para (a,b), (c,d) ∈ A×B, (a,b) = (c,d) ⇔ a = c y b = d.

Para U = {1,2,3,…,7}, A ={2,3,4}, B = {4,5} (*) determine:

- A×B=
- B×A=
- B^2=B×B=
- B^3=

Indique:
- |A| =
- |B| =
- |A×B| =
- |B×A| =

2.2 Relación

Para los conjuntos A, B $\subseteq U$, cualquier subconjunto de A×B es una **relación** de A en B. Cualquier subconjunto de A×A es una **relación binaria** en A.

Para A, B, U, presentado anteriormente (*) las siguientes ¿son relaciones de A en B?

a. ∅
b. {(2,4),(2,5)}
c. {(2,4),(3,4),(4,5)}
d. {(2,3)}
e. {(2,4),(3,4)(4,4)}
f. A×B

En general para conjuntos finitos A, B con |A| = m y |B| = n, existen 2^{mn} relaciones de A en B, incluyendo la relación vacía y la propia relación A×B.

Sea R el subconjunto de N×N donde R = {(m,n) / n = 7m}. En consecuencia, entre los pares ordenados en R se encuentran (0,0), (1,7), (11,77), (15, 105).
Esta relación R en N, también puede representarse en forma recursiva como:

1) (0,0)∈ R
2) Si (s,t) ∈ $\mathfrak{R}$ ⇒ (s + 1, t + 7) ∈ R

Compruebe que el par (3,21) ∈ R

Interpretemos la definición dada para **R**…

Sea **R** el subconjunto de NxN donde **R = {(m,n) / n = 7m}.** Si analizamos un momento la definición de **R** vemos que los pares ordenados van a tener la forma genérica **(m, 7m)** con **m** y **n** naturales. Evidentemente la cantidad de elementos de **R** es infinita, ahora, ¿cuál sería la cuestión? ¿Qué variable uso para recorrer N y por qué tengo que hacer esto? ¿Por qué usar una y no la otra? Lo cierto es que tengo que elegir una y ver que "pasa" con la otra variable. Probemos con m, si a m la multiplicamos por siete (según def. por comprensión de **R**: **n = 7m**) el resultado va a ser un natural cualquiera, por ejemplo n. Si a partir de esto nos armamos una tabla de valores **[m, 7m]** y la completamos con todos los naturales N entonces tendremos la "extensión" de **R**, bueno, a ver, en la práctica, ¿se puede realmente hacer esto? ¡Si y no y por qué!?
¿Qué piensan Uds. de todo esto?

En consecuencia, entre los pares ordenados en R se encuentran (0,0), (1,7), (11,77), (15, 105), …etc. …,
La pregunta que nos hacemos ahora es si será posible expresar lo anterior de una forma más cómoda, digamos algo más práctico.
La respuesta es que si y la forma de resolver esto sería mediante una estructura recursiva como la que se muestra a continuación.

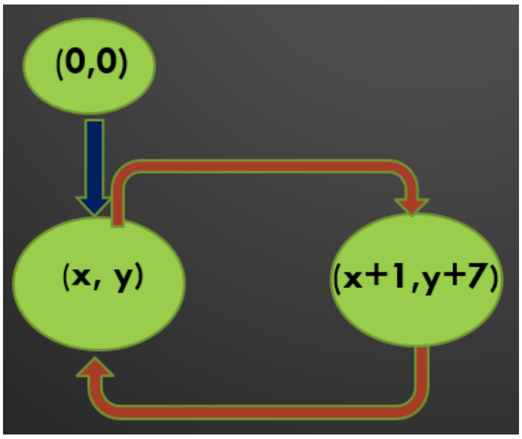
1) $(0,0) \in R$ **[Regla Base]**
2) si $(s,t) \in R \Rightarrow (s + 1, t + 7) \in R$ **[Regla Recursiva]**

¿Y cómo funciona esto realmente?

Siguiendo con el planteo recursivo de R tendríamos lo siguiente:
$(0,0) \in R$ [por Regla Base]
Si $(0,0) \in R \Rightarrow (0 + 1, 0+ 7) \in R$ [por Regla Recursiva]
Pero $(0 + 1, 0+ 7) = (1, 7) \in R$ entonces decimos que:
Si $(1,7) \in R \Rightarrow (1 + 1, 7+ 7) \in R$ y si volvemos a aplicar la Regla Recursiva nuevamente tenemos que:
Si $(2,14) \in R \Rightarrow (2 + 1, 14+ 7) \in R$ y así podríamos seguir indefinidamente!!

Pregunta: ¿Qué ocurrirá cuando esto se continúe una y otra vez? ¿Podremos relacionarlo con la tabla [m, 7m]? Y si fuera así, ¿qué consecuencia tendría esto? ¡Justifique su respuesta!

Si modelamos la relación R en su forma recursiva usando un grafo dirigido y tratamos de verlo como una maquina (y que esta sea de ciclo infinito) tendríamos el siguiente dibujo:



Teorema

Para cualesquiera conjuntos A, B, C $\subseteq U$:

1) $A\times(B\cap C) = (A\times B) \cap (A\times C)$
2) $A\times(B\cup C) = (A\times B) \cup (A\times C)$
3) $(A\cap B)\times C = (A\times C) \cap (B\times C)$
4) $(A\cup B)\times C = (A\times C) \cup (B\times C)$

Consideremos las siguientes Relaciones…

U = {1,2,3,…,7}, A ={1,2,3,4,5,6,7}, B = {1,2,3,4,5,7}

R1 = {(1,2), (2,7), (2,4), (2,5), (4,7)} :: R2 = {(2,5), (2,4), (3,4), (4,5)}

R3 = {(2,3), (2,7), (2,1)} :: R4 = {(2,7), (2,5), (2,4), (3,4), (4,4)}

Resolviendo algunas operaciones entre estas tenemos:

R1 $\cup$ R2 = {(1,2), (2,7), (2,4), (2,5), (4,7), (2,5), (2,4), (3,4), (4,5)}

R1 $\cap$ R2 = {(2,4), (2,5)}

R1 - R2 = {(1,2), (2,7), (4,7)}

R3 $\oplus$ R4 = {(2,3), (2,1), (2,5), (2,4), (3,4), (4,4)}

¡¡Entonces, queda claro que las relaciones son CONJUNTOS y los pares ordenados ELEMENTOS, por consiguiente, aplican las operaciones entre estos!!

2.3 Algunas propiedades de las relaciones

- Una relación R en un conjunto A es **reflexiva** si $\forall$ x $\in$ A, (x,x) $\in$ R.
- Una relación R en un conjunto A es **simétrica** si:
  $\forall$ (x,y) $\in$ R $\Rightarrow$ (y,x) $\in$ R (con x, y $\in$ A)
- Una relación R en un conjunto A es **transitiva** si:
  $\forall$ (x,y), (y,z)$\in$ R $\Rightarrow$ (x,z) $\in$ R (con x, y, z $\in$ A)

Siendo A ={1,2,3} y las relaciones

R_1 ={(1,2), (2,1), (1,3), (3,1)},

R_2 ={(1,1), (2,2),(3,3), (2,3)},

R_3 ={(1,1), (2,2),(3,3)},

R_4 ={(1,1), (2,2),(3,3), (2,3),(3,2)},

R_5 ={(1,1), (2,3),(3,3)}

Determine que propiedades se cumplen en las relaciones anteriores. Justifique

2.4 Relación de equivalencia

- Una **relación de equivalencia** R en un conjunto A es una relación que **es reflexiva, simétrica y transitiva**.

Siendo A ={1,2,3} y las relaciones

R_1 ={(1,1), (2,2),(3,3)},

R_2 ={(1,1), (2,2),(3,3), (2,3),(3,2)},

R_2 ={(1,1), (1,3), (2,2),(3,1), (3,3)},

R_3 = A×A

¿Son las relaciones anteriores, relaciones de equivalencia en A? ¡Justifique!

Ejemplo: Sea A el conjunto de todos los estudiantes de tu escuela y considera la relación R en el conjunto A que consiste en los pares (x, y) tales que x e y tuvieron el mismo promedio de notas el año anterior.

- ¿Es R una relación de equivalencia?
- ¿Dado un estudiante x, podemos formar el conjunto de todos los estudiantes que tuvieron el mismo promedio de notas el año anterior que x?

Se dice que este subconjunto de conjunto A es una clase de equivalencia de la relación R y que x e y son equivalentes.

<u>Solución:</u>

Entonces, a R b si el promedio del sujeto a es igual al promedio del sujeto b.

Letra del ejercicio traducido	a R b sii prom(a) = prom(b)

¡¡IMPORTANTE!! Decir que **(a, b)** pertenece a **R** es lo mismo que decir que **a R b**

<u>¿Es R es de equivalencia?</u> tendrá que cumplir con la *Reflexiva, Simétrica* y *Transitiva*

<u>Reflexiva</u>:
(a,a) pertenece a R es lo mismo que decir que a R a sii prom(a) = prom(a) la cumple!

<u>Simétrica</u>:
(a,b) pertenece a R entonces (b,a) pertenece a R
a R b sii prom(a) = prom(b) entonces b R a prom(b) = prom(a) se cumple!

<u>Transitiva</u>:
(a,b) pertenece a R y (b,c) pertenece a R entonces (a,c) pertenece a R
prom(a) = prom(b) y prom(b) = prom(c) entonces prom(a) = prom(c) entonces a R c
Cumple la transitiva

¡¡Entonces R es de equivalencia!!

2.5 Clases de equivalencia

Sea R una **relación de equivalencia** en un conjunto A. El conjunto de todos los elementos que están relacionados con un elemento a de A se llama **clase de equivalencia** de a. La clase de equivalencia de a con respecto a R se denota **$[a]_{\mathfrak{R}}$**.

Si $b \in [a]_{\mathfrak{R}}$, se dice que b es un **representante** de esta clase de equivalencia. Cualquier elemento de la clase se puede usar como representante de dicha clase.

De acuerdo a la relación R del último ejemplo. ¿Cuántos subconjuntos de A puedo tener? ¿Esos conjuntos son disjuntos?

Pero antes de contestar esto preguntémonos…

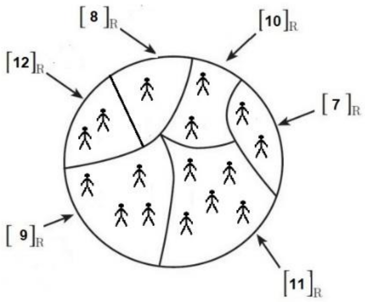
¿Que genera una *Relación de equivalencia* sobre el conjunto A?

Bien, lo que genera es una partición especial del conjunto sobre el que se define la relación binaria. A cada conjunto de los que integran la partición le llamamos clase de equivalencia y en cada clase encontramos elementos que se dicen equivalentes entre sí.

En resumen, diremos que siendo R una relación de equivalencia sobre un conjunto A, para cualquier $x \in A$ la clase de equivalencia de x se define como:

$$[X] = \{y \in A |\ (y, x) \in R\}$$

Para el caso del ejemplo teórico la cuestión sería más o menos así…



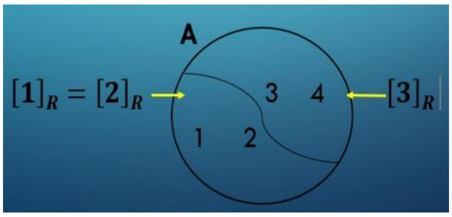
Podemos observar las distintas clases de Equivalencia (que son subconjuntos de A), con "representante" 8, 12, 10, etc.., en cada una de estas clases están todos los individuos que el año anterior pasaron con el mismo promedio de notas.

Nótese que la unión de todas las clases me da el conjunto A de alumnos de alguna escuela UTU.

Veamos un par de ejemplos más…

Ejemplo 1: Sea A = {1,2,3,4} y R una relación de A en A tal que:

R={(1,1),(1,2),(2,1),(2,2),(3,3),(3,4),(4,3),(4,4)} **(*)**



Se pide comprobar que es de equivalencia y determinar las clases.
Vemos que efectivamente es de equivalencia (¡cumple con la Reflexiva, Simétrica y Transitiva!) y aplicando la definición de Clase de equivalencia tenemos que:

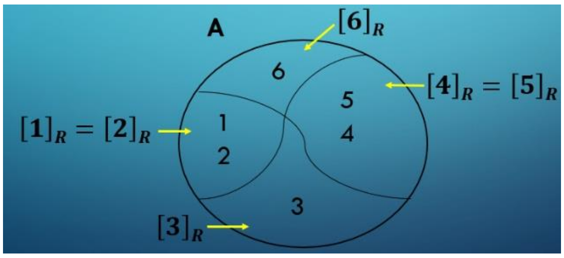
[1]={1,2}; [2]={1,2} *y* [3]={3,4} si representamos estas clases en un diagrama tendríamos lo siguiente **(*)**.

Ejemplo 2: Sea A = {1,2,3,4,5,6} y R una relación de A en A tal que:

R={(1,1),(1,2),(2,1),(2,2),(3,3),(4,4),(4,5),(5,4),(5,5),(6,6)}

Se pide comprobar que es de equivalencia y determinar las clases.
¡Ídem anterior! (¡cumple con la Reflexiva, Simétrica y Transitiva!) y aplicando la definición de Clase de equivalencia tenemos que:

[1]={1,2}; [2]={1,2} ; [3]={3} ; [4]={4,5}; [5]={4,5} ; [6]={6}



2.6 Representación de relaciones: Matrices y Grafos dirigidos

Hay muchas formas de representar una relación entre conjuntos finitos. Como ya henos visto, una de ellas es **enumerar los pares ordenados**. Presentaremos dos métodos alternativos, para representar relaciones. Un método utiliza **matrices booleanas**. El otro método hace uso de **grafos dirigidos**.

Por lo general, las matrices son estructuras apropiadas para las representaciones de relaciones en programas informáticos. Por otra parte, muchas veces resulta útil representar las relaciones mediante grafos dirigidos para entender las propiedades de dicha relación.

2.7 Matriz

Una **matriz** es una **disposición rectangular de números**. Una matriz con m filas y n columnas se denomina **matriz de m×n**. Una matriz con el **mismo número de filas que de columnas** se denomina **matriz cuadrada**. Dos matrices son iguales si tienen el mismo número de filas que de columnas y los elementos en cada posición son iguales dos a dos.

$$M_1=\begin{pmatrix}1 & 1\\ 0 & 2\\ 1 & 3\end{pmatrix} \qquad M_2=\begin{pmatrix}1 & 3 & 2 & 0\\ 1 & 0 & 0 & 1\\ 2 & 1 & 4 & 1\end{pmatrix} \qquad M_3=\begin{pmatrix}1 & 0\\ 0 & 0\end{pmatrix}$$

¿Podría indicar las dimensiones de cada una de las matrices anteriores?

$M_1 =$? $\qquad M_2 =$? $\qquad M_3 =$?

2.7.1 Representación de relaciones usando matrices

Una **relación** entre conjuntos finitos **se puede representar** utilizando una **matriz booleana**. Supongamos que R es una relación de A= $\{a_1, a_2, a_3,\ldots\ldots, a_m\}$ en B = $\{b_1, b_2, b_3,\ldots\ldots, b_n\}$ (escribimos los elementos de los conjuntos A y B en un orden particular aunque arbitrario. Además, cuando A= B usamos la misma ordenación para A y para B). la relación R puede representarse por medio de la matriz $M_R=[m_{ij}]$, donde:

$$m_{ij}=\begin{cases} 1 \text{ si } (a_i,b_j) \in R \\ 0 \text{ si } (a_i,b_j) \notin R \end{cases}$$

En otras palabras, la matriz booleana que representa a R tiene un 1 como elemento (i,j) si a_i está relacionado con b_j y tiene un 0 en esta posición si a_i no está relacionado con b_j.

Supongamos A = {1,2,3} y B = {1,2}. Sea R la relación de A en B que contiene a (a,b) si a $\in$ A, b $\in$ B y a > b. Si $a_1 = 1$, $a_2 = 2$, $a_3 = 3$, y $b_1 = 1$, $a_2 = 2$ obtenemos la relación siguiente R ={(2,1), (3,1),(3,2)} por tanto la matriz que representa dicha relación es:

$$M_{\Re} = \begin{pmatrix}0 & 0\\ 1 & 0\\ 1 & 1\end{pmatrix}$$

Los unos en $M_{\Re}$ muestran que sólo los pares (2,1), (3,1) y (3,2) pertenecen a R.

Sean A= $\{a_1, a_2, a_3\}$ en B = $\{b_1, b_2, b_3, b_4, b_5\}$ ¿Qué pares ordenados están en la relación R representada por la matriz $M_{\Re}$?

$$M_{\Re} = \begin{pmatrix}0 & 1 & 0 & 0 & 0\\ 1 & 0 & 1 & 1 & 0\\ 1 & 0 & 1 & 0 & 1\end{pmatrix} \quad \Rightarrow R = \rule{10cm}{0.4pt}$$

Las matrices de una relación binaria, que son cuadradas, pueden usarse para determinar si la relación cumple o no ciertas propiedades.

Recordemos

- Una relación R en un conjunto A es **reflexiva** si $\forall\ x \in A, (x,x) \in R$.
  R es **reflexiva** $\Leftrightarrow (a_i,a_i) \in R$ para $i = 1,2,3,\ldots\ldots,n$
  Por consiguiente
  R es **reflexiva** $\Leftrightarrow m_{ii} = 1$ para $i = 1,2,3,\ldots\ldots,n$

$$M_{\Re} = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}$$

- Una relación R en un conjunto A es **simétrica** si $(x,y) \in R \Rightarrow (y,x) \in R$ $\forall\ x, y \in A$
  R es **simétrica** $\Leftrightarrow (a_j,a_i) \in R$ sempre que $(a_i,a_j) \in R$
  Por consiguiente
  R es **simétrica** $\Leftrightarrow m_{ji} = 1$ siempre que $m_{ij} = 1$
  (esto significa que $m_{ji} = 0$ siempre que $m_{ij} = 0$)

$$M_{\Re} = \begin{pmatrix} & & 1 & & \\ & & & & \\ 1 & & & & 0 \\ & & & & \\ & & 0 & & \end{pmatrix}$$

Dadas las siguientes relaciones representadas en su forma matricial, ¿podría indicar si cumplen las propiedades simétricas y/o reflexiva?

$$M_{\Re_1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \qquad M_{\Re_2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

$$M_{\Re_3} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \qquad M_{\Re_4} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

R_1 ____

R_2 ____

R_3 ____

R_4 ____

Unión e intersección de matrices booleanas

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

La unión de las matrices A y B se denota $A \vee B$

$$A \vee B = \begin{pmatrix} 1\vee 0 & 0\vee 1 & 1\vee 0 \\ 0\vee 1 & 1\vee 1 & 0\vee 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

La intersección de las matrices A y B se denota $A \wedge B$

$$A \wedge B = \begin{pmatrix} 1\wedge 0 & 0\wedge 1 & 1\wedge 0 \\ 0\wedge 1 & 1\wedge 1 & 0\wedge 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Las operaciones de unión e intersección de matrices nos pueden ayudar para determinar las matrices que representan la unión y la intersección de dos relaciones.

$$M_{\mathcal{R}_1 \cup \mathcal{R}_2} = M_{\mathcal{R}_1} \vee M_{\mathcal{R}_2}$$

La matriz que representa la unión de dos relaciones es la unión de las matrices que representan las relaciones

$$M_{\mathcal{R}_1 \cap \mathcal{R}_2} = M_{\mathcal{R}_1} \wedge M_{\mathcal{R}_2}$$

La matriz que representa la intersección de dos relaciones es la intersección de las matrices que representan las relaciones.

Supongamos que las relaciones R_1 y R_2 en un conjunto A están representadas por las matrices.

$$M_{\mathcal{R}_1} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \qquad M_{\mathcal{R}_2} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

¿Cuáles son las matrices que representan a $R_1 \cap R_2$ y $R_1 \cup R_2$?

2.7.2 Representación de relaciones usando grafos dirigidos

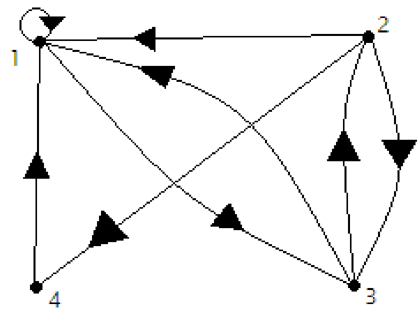
Hemos visto que una relación se puede representar enumerando todos sus pares ordenados o utilizando una matriz booleana. Hay otra manera de representar una relación por medio de una representación gráfica. Cada elemento del conjunto se representa mediante un punto y cada par ordenado se representa mediante un segmento orientado cuyo sentido viene indicado por una flecha. Utilizamos esta representación gráfica cuando pensamos en las relaciones en un conjunto finito cono grafos dirigidos.

2.8 Grafo dirigido

Un **grafo dirigido** consta de un **conjunto V de vértices** (o nodos) junto con un **conjunto E de pares ordenados** de elementos de V llamados aristas (o arcos). Al vértice a se le llama vértice inicial de la arista (a,b) y al vértice b se le llama vértice final de esta arista.

Una **arista (a,a)** se representa usando un arco que conecta el vértice a consigo mismo, una arista de esta forma se llama **bucle**.

Dada la relación $\mathcal{R}$ ={(1,1), (1,3),(2,1),(2,3),(2,4),(3,1),(3,2),(4,1)} en el conjunto A={1,2,3,4} está representado por el siguiente grafo dirigido.



El grafo dirigido que representa una relación puede utilizarse para determinar si la relación cumple o no diversas propiedades.

– Una relación $\mathcal{R}$ en un conjunto A es **reflexiva** si $\forall$ x $\in$ A, (x,x) $\in$ $\mathcal{R}$.

$\mathcal{R}$ es **reflexiva** si y solo si hay un bucle en cada vértice del grafo dirigido, de modo que todos los pares ordenados de la forma (x,x) pertenecen a la relación.

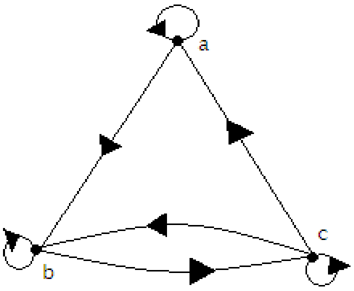
– Una relación $\mathcal{R}$ en un conjunto A es **simétrica** si $\forall$ (x,y) $\in$ $\mathcal{R} \Rightarrow$ (y,x) $\in$ $\mathcal{R}$ con x, y $\in$ A

$\mathcal{R}$ es **simétrica** si para cada arista entre vértices distintos del grafo dirigido existe una arista en sentido opuesto, de modo que (y, x) está en la relación siempre (x, y) lo está.

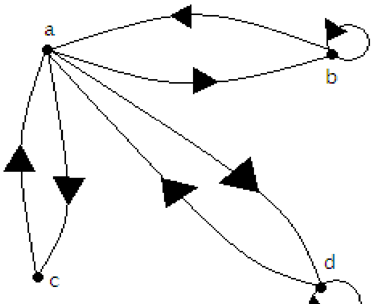
– Una relación $\mathcal{R}$ en un conjunto A es **transitiva** si:
  $\forall$ (x,y), (y,z)$\in$ $\mathcal{R} \Rightarrow$ (x,z) $\in$ $\mathcal{R}$ con x, y, z $\in$ A

R es **transitiva** si y solo si, siempre que hay una arista uniendo un vértice x con un vértice y una arista uniendo el vértice y con el vértice z, hay una tercera arista que une x con z (completando un triángulo en el que cada lado es una arista orientada en la dirección correcta).

Supongamos que las relaciones R_1 en un conjunto A y R_2 en un conjunto B están representadas por los grafos que se muestran a continuación. Determine los conjuntos A y B. ¿R_1 y R_2 son reflexivas, simétricas y/o transitivas?



Grafo dirigido de $\Re_1$



Grafo dirigido de $\Re_2$

<u>Inversa de una Relación</u>

Para los conjuntos A, B $\subseteq$ U, si R es una relación de A en B, entonces la inversa de R la anotamos R^{c}, es la relación de B en A definida por $R^{c} = \{(b,a) \mid (a,b) \in R\}$

Supongamos A = {1,2,3} y B = {0,1,2}. Sea R_1 ={(1.0), (1,2),(2,1),(2,2), (3,0),(3,1)} y R_2 la relación a relación de A en B que contiene a (a,b) si a$\in$A, b$\in$B y a$\leq$b

- Determine R_2
- Halle R_1^{c} y R_2^{c}
- Represente R_1, R_2, R_1^{c} y R_2^{c} mediante grafos dirigidos.

<u>2.9 Práctico de Relaciones</u>

1. Si U = N, A ={1,2,3,4}, B = {2,5} y C ={3,4,7} determine
   a. A×B
   b. B×A
   c. A∪(B×C)
   d. (A∪B)×C
   e. (A×C) ∪ (B×C)

2. Si U = {1,2,3,4,5}, A ={1,2,3} y B = {2,4,5} encuentre ejemplos

a. Tres relaciones no vacías de A en B
b. Tres relaciones binarias no vacías en A

3. Si A ={1,2,4,8,16} y B = {1,2,3,4,5,6,7}. Si (2 – x,5) (4, y – 2) ∈ A×B ¿Se cumple que (2 – x,5) = (4, y – 2)?

4. Para A, B, U los del ejercicio 2 determine

a. |A×B|
b. El número de relaciones binarias en A

5. ¿Para cuales conjuntos A, B ⊆ U es verdadero que A×B = B×A?

6. Para un universo dado U, sean A, B, C, D subconjuntos no vacíos de U.

a. Demuestre que A×B ⊆ C×D ⇔ A ⊆ C y B ⊆ D
b. ¿Qué le sucede al resultado de la parte a. si cualquiera de los conjuntos A, B, C, D es vacío?

7. Se **R*** la relación definida por **x** + **y** < **7.**
Esto es **x R* y** si y solo si **x** + **y** < **7** y $R* = \{\langle x,y\rangle \in SxS / x + y < 7\}$ *Ptos.*
Para $S = \{1,2,4\}$ contestar lo siguientes:

a) Determinar **SxS.**
b) Dar **R*** por extensión**.**
c) Cumple **R*** con la *Simétrica*? (justifique)
d) Determinar el grafo de **R***.

8. Sea R ⊆ N×N donde (m,n) ∈ R ⇔ n = 5m + 2

a. Plantee una definición recursiva para R
b. Utilizando la definición recursiva de la parte a., muestre que (4,22) ∈ R

9. Sea A={a, b, c, d}, dar ejemplos de relaciones sobre A que sean:

a) Reflexiva y simétrica, pero no transitiva.

b) Reflexiva y transitiva, pero no simétrica.

c) Simétrica y transitiva, pero no reflexiva.

10. Para cada una de las siguientes relaciones, determine si es **reflexiva**, **simétrica** o **transitiva**.

a) R⊆N*x N*, definida como *aRb*, si $a|b$ (*a* divide a *b* o *a* es divisor de *b*).

b) R es la relación sobre Z tal que *xRy* si *x+y* es un número: i) par ii) impar.

c) R es la relación sobre Z tal que *xRy* si *x- y* es un número i) par ii) impar.

d) R es la relación sobre el conjunto N, definida por *aRb* si *a*≤*b*.

e) R es la relación sobre el conjunto Z*, definida por *aRb* si *ab*≥0.

11. Dado U = {1,2,3,4,5}, A ={1,2,3} y B = {2,4,5} con A, B ⊆ U contestar lo que sigue:

a) R_1 ={(1,1), (2,2), (3,3)} es una relación Reflexiva. **V** **F**

b) R_2 ={(1,1), (2,2), (3,3), (2,3), (3,2)} es una relación de Equivalencia. **V** **F**

c) R_3 ={(2,3), (3,1), (2,2)} es una relación Transitiva. **V** **F**

d) R_4 = A×A es una relación vacía. **V** **F**

e) R_5 ={(1,2), (1,5), (2,2), (3,4), (3,5)} es una relación de A en B. **V** **F**

f) R_6 ={(5,1), (1,5), (4,4), (3,1), (3,3)} es una relación de B en A. **V** **F**

g) R_7 ={(5,5), (4,4), (2,4)} es una relación de B^2. **V** **F**

h) R_8 ={(1,2), (1,4), (2,2), (3,4), (3,5)} es una relación $R\colon A \rightarrow B$. **V** **F**

12. Sea A ={1, 2, 3, 4} y la relación R1 tal que:
$R1$ ={(1, 1), (1, 2), (2, 1), (2, 2), (3, 3), (3, 4), (4, 3), (4, 4)}
a) Comprueba que R es una relación de equivalencia.
b) Representa el grafo dirigido de R y halla las clases de equivalencia.
c) ¿Cuál es la partición que induce R sobre A?

13. Considera sobre el subconjunto D = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} de números enteros, la relación de equivalencia R definida como aRb si y sólo si $a–b$ es múltiplo de 4.

a) Determina la relación R.
b) Halla las clases de equivalencia.

14. ¿Cuáles de estas relaciones en el conjunto {0,1,2,3} son relaciones de equivalencia? Determine las propiedades que le faltan a las restantes para ser relación de equivalencia.

c. R_1 ={(0,0), (1,1), (2,2),(3,3)}
d. R_2 ={(0,0), (0,2), (2,0),(2,2), (2,3), (3,2) (3,3)}
e. R_3 ={(0,0), (1,1), (1,2), (2,1),(2,2),(3,3)}
f. R_4 ={(0,0), (1,1), (1,3), (2,2),(2,3), (3,1), (3,2), (3,3)}
g. R_5 ={(0,0), (0,1), (0,2), (1,0), (1,1), (1,2),(2,0), (2,2), (3,3)}

15. ¿Cuáles de estas relaciones en el conjunto de todas las personas son relaciones de equivalencia? Determine las propiedades que le faltan a las restantes para ser relación de equivalencia.

h. {(a,b) / a y b tienen la misma edad}
i. {(a,b) / a y b tienen los mismos padres}
j. {(a,b) / a y b tienen uno de los padres en común}
k. {(a,b) / a y b se conocen}
l. {(a,b) / a y b hablan un mismo idioma}

16. Demuestre que una relación R, que consiste en todos los pares (x,y) en los que x e y son cadenas de bits de longitud al menos 3 coincidan en sus tres primeros bits es una relación de equivalencia en cadenas de bits de longitud al menos 3

17. Sea $\mathcal{R}$ la relación en el conjunto de pares ordenados de enteros positivos tal que $((a,b),(c,d)) \in \mathcal{R} \Leftrightarrow ad = bc$. Demuestre que $\mathcal{R}$ es una relación de equivalencia.

18. Represente cada una de estas relaciones en el conjunto A={1,2,3} mediante una matriz (con los elementos del conjunto A en orden creciente)

m. $\mathcal{R}_1 =\{(1,1), (1,2),(1,3)\}$

n. $\mathcal{R}_2 =\{(1,2), (2,1),(2,2),(3,3)\}$

o. $\mathcal{R}_3 =\{(1,1), (1,2), (1,3),(2,2),(2,3),(3,3)\}$

p. $\mathcal{R}_4 =\{(1,3), (3,1)\}$

19. Represente las relaciones en el conjunto A={1,2,3} que están representadas por medio de las siguientes matrices (las filas y las columnas de las matrices corresponden a los enteros escritos en orden creciente) mediante pares ordenados.

q. $M_{\mathcal{R}_1} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ r. $M_{\mathcal{R}_2} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ s. $M_{\mathcal{R}_3} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

20. Represente las relaciones en el conjunto A={1,2,3,4} que están representadas por medio de las siguientes matrices (las filas y las columnas de las matrices corresponden a los enteros escritos en orden creciente) mediante pares ordenados.

t. $M_{\mathcal{R}_1} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$ u. $M_{\mathcal{R}_2} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{pmatrix}$ v. $M_{\mathcal{R}_3} = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$

21. Determine si las relaciones de los ejercicios 13 y 14 son reflexivas, simétricas, y/o transitivas.

22. Sean $\mathcal{R}_1$ y $\mathcal{R}_2$ relaciones en un conjunto A representadas por las matrices

$M_{\mathcal{R}_1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ $M_{\mathcal{R}_1} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$

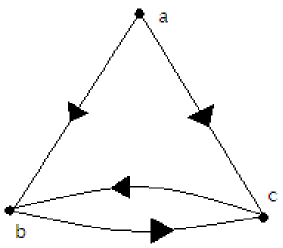
w. Halle la matriz que representa $\mathcal{R}_1 \cap \mathcal{R}_2$

x. Halle la matriz que representa $\mathcal{R}_1 \cup \mathcal{R}_2$

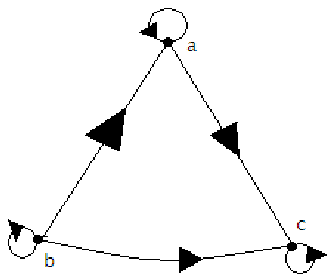
23. Dibuje los grafos dirigidos que representan a las relaciones del ejercicio 14.

24. Dibuje el grafo dirigido que representa la relación $\mathcal{R}$ en el conjunto A={a,b,c,d}, siendo $\mathcal{R} =\{(a,a), (a,b), (b,c), (c,b),(c,d), (d,a), (d,b)\}$ Determine R^c

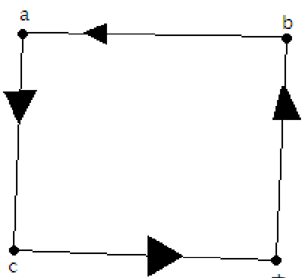
25. Dados los siguientes grafos que representan relaciones, indique el conjunto sobre el cual está definida la relación, represéntela por medio de pares ordenados e indique si son reflexivas, simétricas y/o transitivas. ¿Hay alguna que sea relación de equivalencia?



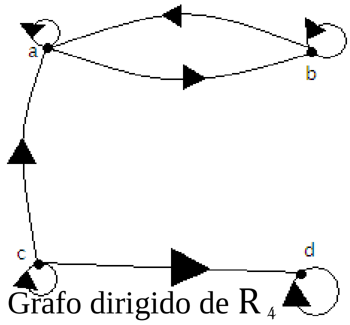
Grafo dirigido de R_1



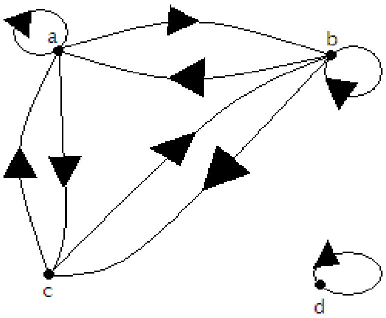
Grafo dirigido de R_2



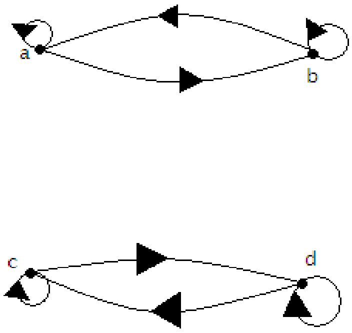
Grafo dirigido de R_3



Grafo dirigido de R_4



Grafo dirigido de R_5



Grafo dirigido de R_6

26. Sea $\Re$ la relación en el conjunto de los números reales tal que

$(x,y) \in \Re \Leftrightarrow x - y = x^3 - y^3$.

y. Demuestre que R es una relación de equivalencia.
z. Halle las clases de equivalencia de los reales 1, 2 y 3.

27. Sea R la relación en el conjunto de pares ordenados de enteros tal que $((x,y),(x',y')) \in \mathrm{R} \Leftrightarrow x - x' = 2(y - y')$.

aa. Demuestre que R es una relación de equivalencia.

bb. Halle las clases de equivalencia de (0, 0) y de (2, -1).

3 Funciones

3.1 Definiciones

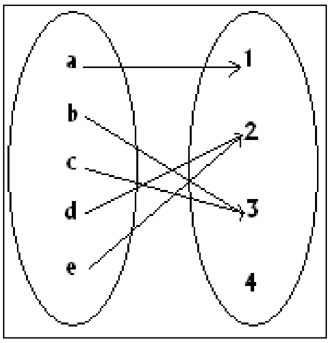
D1: Se dice que una relación **R** de **A** en **B** es una *función* si para cualquier elemento **a** en **A**, existe un único elemento **b** de **B** tal que (a,b) está en **R**.

D2: Para los conjuntos no vacíos **A**, **B**, una *función*, o aplicación **f** de **A** en **B**, que se denota con $f : A \to B$, es una relación de **A** en **B** en la que cada elemento de **A** aparece exactamente una vez como la primera componente de un par ordenado de la relación.

Con frecuencia escribimos f(a) = b cuando (a,b) es un par ordenado en la *función* **f**. Si (a,b) pertenece a **f**, **b** se conoce como la imagen de mediante **f**, mientras que **a** es una preimagen de **b**. *Además, la definición sugiere que **f** es un método para asociar a cada **a** que pertenece a **A** una única **b** que pertenece a **B**, anotamos este proceso como f(a) = b.* En consecuencia, decir que (a,b) y (a,c) pertenecen a **f** implica necesariamente que **b = c**.

A menudo una función puede ser representada en forma gráfica. La figura 1 a) muestra una función R de A={a,b,c,d,e} en B={1,2,3,4}. Si seguimos la convención de representar una relación binaria en forma tabular, podemos representar la función de la figura 1 a) como la de la figura 1 b). No obstante, una forma tabular más conveniente para representar funciones es la mostrada en la figura 1 c), donde la columna izquierda contiene todos los elementos en el dominio y la columna derecha contiene sus imágenes correspondientes.

Figura 1



a)

f(x)	1	2	3	4
a	x			
b			x	
c			x	
d		x		
e		x		

b)

x	f(x)
a	1
b	3
c	3
d	2
e	2

c)

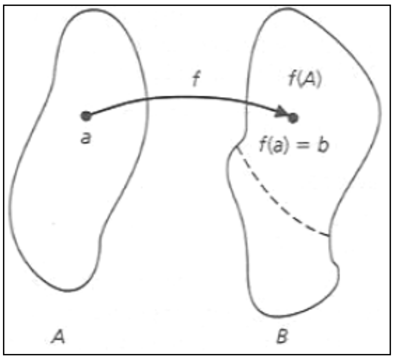
Ejemplo: Sean $A = \{1,2,3\}$ y $B = \{w,x,y,z\}$, y $R_1 = \{(1,w),(2,x),(3,x)\}$, $R_2 = \{(1,w),(2,x)\}$ y $R_3 = \{(1,w),(2,w),(2,x),(3,z)\}$ ¿Son R_1, R_2 y R_3 relaciones de A en B? ¿Alguna de ellas es función? Justifique.

3.2 Dominio, Codominio y Conjunto Imagen

D3: Para la función $f : A \to B$, **A** es el dominio de **f** y **B** es el codominio de **f**. El subconjunto de **B** formado por aquellos elementos que aparecen como segundas componentes de los pares ordenados de **f** se conoce como la imagen de **f** y se denota también como **f(A)** ya que es el conjunto de imágenes (de los elementos de **A**) mediante **f**. En otras palabras: $\mathrm{Im}(f) = \{f(x) : x \in Dom(f)\}$

Una representación gráfica de estas ideas aparece en la figura 2. Este diagrama sugiere que **a** puede verse como *entrada* que es *transformada* por **f** en la *salida* correspondiente **f(a).**

Figura 2

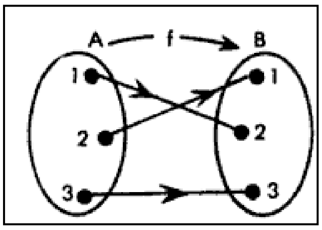


3.3 Función Inyectiva

Una función $f : A \to B$ se denomina *inyectiva*, o *uno a uno* si distintos elementos de **A** tienen distintas imágenes en **B** bajo **f**:
t.q. $x_1, x_2 \in A$ y $x_1 \neq x_2$ implica $f(x_1) \neq f(x_2)$
Esto equivale lógicamente a la contrapositiva: $x_1, x_2 \in A$ y $f(x_1) = f(x_2)$ implica $x_1 = x_2$.

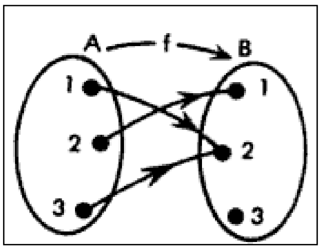
Ejemplo 1: Sea A={1,2,3} B={1,2,3}; f: A→B: f={(1,2), (2,1), (3,3)}
¿f es inyectiva? ¿justifique?

Figura 3



Ejemplo 2: Sea A={1,2,3} B={1,2,3}; f: A→B: f={(1,2), (2,1), (3,2)}
¿F es inyectiva? ¿Por qué? Explique.

Figura 4



Ejemplo 3: Para la siguiente función: $f(x) = y = x - 1$
¿Qué dominio tiene esta función?, ¿Qué tipo de función es f? ¡Justifique su respuesta!

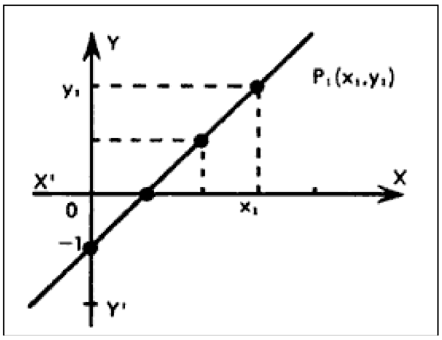


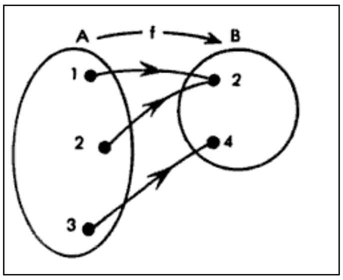
Figura 5

3.4 Función Sobreyectiva

Dada $f : A \to B$ decimos que f es ***sobre**yectiva*, o que **f** transforma a **A *sobre* B** si **Img(f) = B.** En otras palabras $\Leftrightarrow\ \forall y \in B, \exists x \in A \mid f(x) = y$

Ejemplo 4: Sean los conjuntos:
A = {1,2,3} y B = {2,4} y la función f = {(1,2), (2,2), (3,4)}

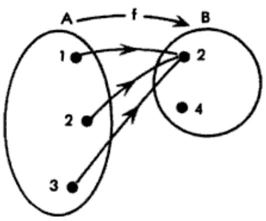
Figura 6



¿Cuál es el conjunto imagen de f?
¿Cuál es el codominio?
¿Qué puedo decir del conjunto Imagen y del codominio?
¿f es sobreyectiva?

Ejemplo 5: Para los mismos conjuntos del ejemplo anterior con la función:
f = {(1,2), (2,2), (3,2)}:: ¿Puede asegurar que f es sobreyectiva? ¡Justifique!

Figura 7

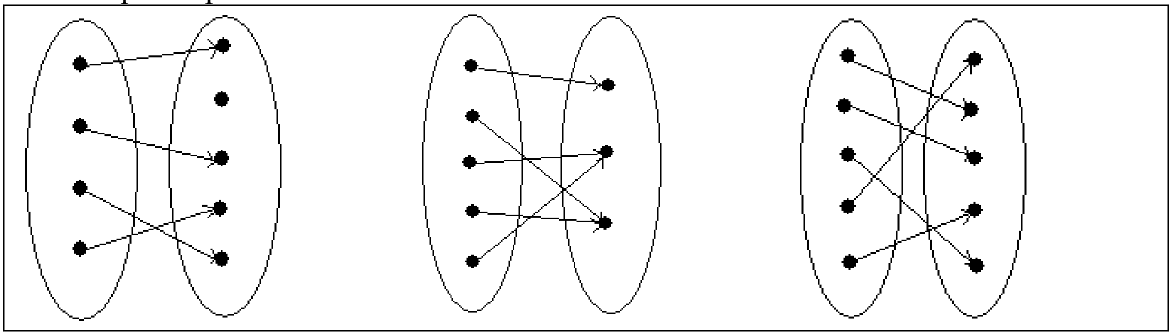


3.5 Función Biyectiva

D6: Una función $f : A \to B$ es *biyectiva* si es inyectiva y *sobreyectiva* a la vez. O lo que es lo mismo: $\forall y \in B, \exists! x \in A |\ f(x) = y$

¿Cuál de los ejemplos anteriores cumple con esta definición? ¿Por qué? ¡Justifique!

Otro esquema posible …

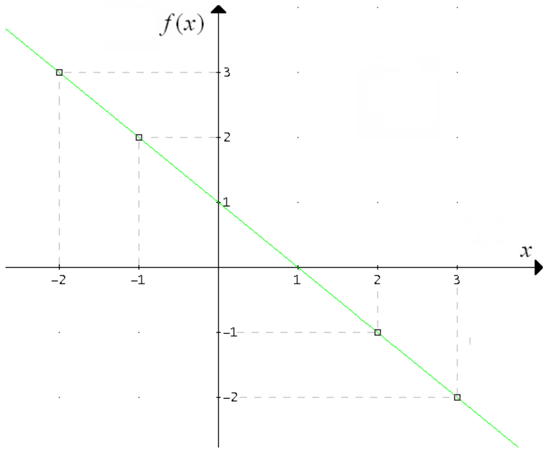


Función Inyectiva *Función Sobreyectiva* *Función Biyectiva*

3.6 Gráfico de una función

Sea la función $f : R \to R / f(x) = -x + 1$. Complete la tabla para diferentes valores de x.

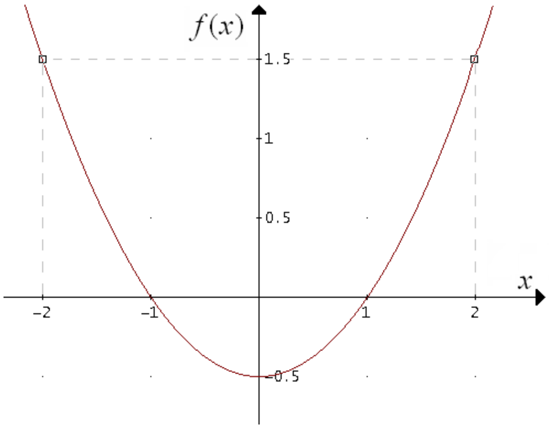
x	$f(x)$
-2	
-1	
0	
1	
2	
3	



Observando el gráfico, ¿Se puede determinar si f es inyectiva y/o sobreyectiva?

Sea ahora la función $f: R \to R \big/ f(x) = \frac{1}{2}x^2 - \frac{1}{2}$. Complete la tabla para diferentes valores de x.

x	$f(x)$
-2	
-1	
0	
1	
2	



Observando el gráfico, ¿Se puede determinar si f es inyectiva y/o sobreyectiva?

Sean las funciones $f: R \to R$ y g:R $\to R \ \big/ \ f(x) = 5x - 3,\ g(x) = x^2$

- Grafique las funciones f y g
- ¿Puede indicar si f, g son inyectivas o sobreyectivas?

3.7 Función Identidad

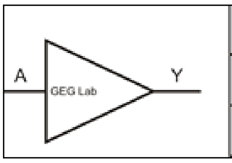
D7: La función $I_A : A \to A$, definida como $I_A(a) = a \ \forall a \in A$, es la *función identidad* para **A**.

3.8 Igualdad de Funciones

D8: Si $f, g : A \to B$, decimos que **f** y **g** son iguales y escribimos f = g si f(a) = g(a) para todo **a** que pertenezca a **A**.

D8.1 Dadas dos funciones f : A ⟶ B y g : C ⟶ D, son iguales o idénticas si se cumple:
Tienen el mismo dominio: A = C
Tienen el mismo codominio: B = D
Asignan las mismas imágenes: para cada x ∈ A = C, se tiene que f(x) = g(x)

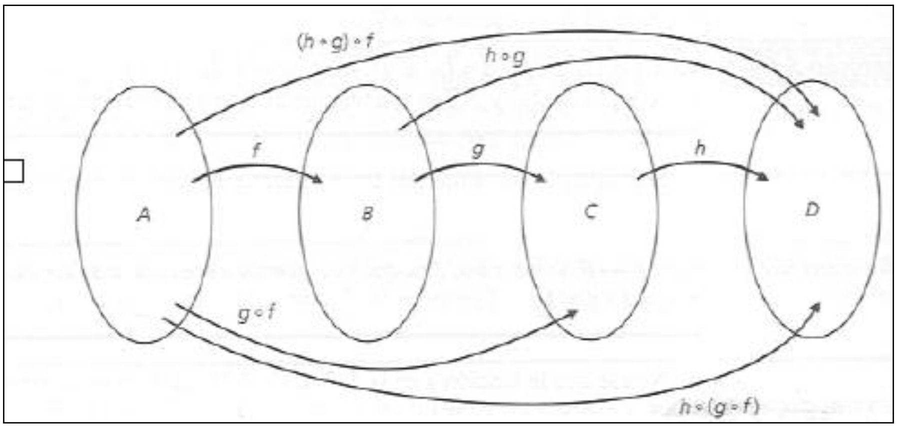
Un ejemplo con puertas lógicas
La función igualdad es aquella función en cuya salida (Y) existe el mismo valor que en su entrada (A).
Su tabla de verdad y símbolo serían los siguientes:



A	Y
0	0
1	1

3.9 Función Compuesta

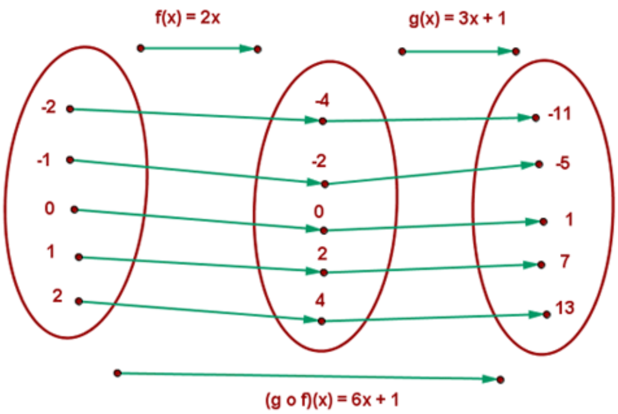
D9: Si $f : A \to B$ y $g : B \to C$, definimos la *función compuesta*, que se denota $g \circ f ; A \to C$, como $(g \circ f)(a) = g(f(a)) \quad \forall a \in A$.



La definición de composición de funciones requiere que el *codominio de f = dominio de g*. Si la imagen de **f** $\subseteq$ *dominio de g*, en realidad esto será suficiente para obtener la composición de $g \circ f ; A \to C$.

Ejemplo 6:

Dadas f(x) y g(x), tal que el dominio de la 2ª esté incluido en el recorrido de la 1ª, se puede definir una nueva función que asocie a cada elemento del dominio de f(x) el valor de g[f(x)].



(g o f) (x) = g [f(x)] = g (2x) = 3 (2x) +1 = **6x + 1**

(g o f) (1) = 6 · 1 + 1 = 7

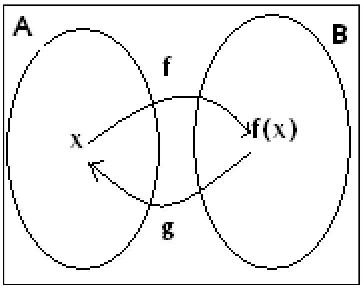
Sea f: R → R, g: R → R definidas por f(x) = x^2, g(x) = x+5

- Grafique f(x), b(x)
- Determine (g o f) (x) y (f o g) (x)

3.10 Función Inversa

D10: Si $f : A \to B$, entonces se dice que f es *invertible* si existe una función $g : B \to A$ tal que $g \circ f = I_A$ y $f \circ g = I_B$.

Teorema: Consideremos una función $f : A \to B$. La función **f** es *invertible* si y solo si **f** es *inyectiva* y *sobreyectiva* a la vez.



- *Como consecuencia de esto, solo podremos hallar la inversa de una función si esta es biyectiva.*

Ejemplo 7:
Para m, b $\in$ **R**, m $\neq$ 0, la función f: R → R definida por $f = \{(x,y) | y = mx + b\}$ ¿f es una función invertible?

Para obtener f^{-1} notamos que:

$$f^{-1} = \{(x,y) | y = mx + b\}^c = \{(y,x) | y = mx + b\}$$
$$= \{(x,y) | x = my + b\} = \{(x,y) | y = (1/m)(x-b)\}$$

Así, f: R→R está definida por $y = mx + b$, y $f^{-1}(x) = (1/m)(x-b)$

Sea $f : Z \to Z \,/\, f(x) = x + 1$. ¿Es invertible? Si lo es, ¿cuál es su inversa?

Sea $g : Z \to Z \,/\, g(x) = x^2$. ¿Es invertible? Si lo es, ¿cuál es su inversa?

3.11 Conjunto PreImagen

D11: Si $f : A \to B$ y $B_1 \subseteq B$ entonces $f^{-1}(B_1) = \{x \in A | f(x) \in B_1\}$. Al conjunto $f^{-1}(B_1)$ se le conoce como la *preimagen de B_1 mediante f.*

¡Cuidado! Aunque tengamos el concepto de una preimagen para cualquier función, no toda función tiene una función inversa. En consecuencia, no podemos suponer la existencia de una inversa para una función f solo porque se usa el símbolo f^{-1}. En este caso, se necesita un poco de precaución.

3.12 Práctico de Funciones

1) Sean A = {a, b, c} B = {1, 2, 3}
   a) De las siguientes relaciones de A en B ¿cuáles son funciones?
      R_1= {(a, 1), (b, 2), (c, 3)}
      R_2= {(a, 1), (b, 1), (c, 1)}
      R_3= {(a, 3), (a, 2), (b, 2), (c, 2)}
      R_4= {(a, 3), (b, 2), (c, 2)}
      R_5= {(b, 2), (c, 2)}
      R_6= {(b, 2), (b, 2), (c, 3)}

b) En caso de ser función determine el conjunto imagen.
c) Haga el diagrama de flecha para las relaciones del ejercicio anterior.
d) Determine un método a través del diagrama para determinar si es o no es función.

2) Sean A = {1,2,3,4}, B = { 1,2,3,4,5,6,7,8} y C = {1,2,3,4,5,6}
Sea f:A→B tal que f = { (1,2), (2, 4), (3, 6), (4, 8) }
Sea g:A→B tal que f(x) = 2.x
Sea h:A→C tal que h(x) = 2.x
a) Determine si ¿f = g? ¿f = h? ¿g = h?
b) Determine para cada función el dominio, el codominio y la imagen.

3) Clasifique las siguientes funciones como inyectivas, sobreyectivas y biyectivas:
a) Sean A = {1,2,3} B = {a,b} Sea f:A→B tal que f = {(1,a), (2, a), (3, b)}
b) Sean A = {1,2,3} B = {a,b,c} Sea g:A→B tal que g = {(1,a), (2,a), (3,a)}
c) Sean A = {1,2,3} B = {a,b,c} Sea h:A→B tal que h = {(1,a), (2,b), (3,c)}
d) Sean A = {1,2} B = {a,b,c} Sea i:A→B tal que i = {(1,a), (2,b)}

4) Sea P = conjunto de las personas del ITS Sean las siguientes funciones:
Cédula: P →{1,...,10000000} tal que *Cédula*(x) = el número de CI correspondiente a la persona x
Edad: P→{1,...,100} tal que *Edad*(x) = el número correspondiente a la edad de x
Altura:P →{1,...,1000} tal que *Altura*(x) = el número que corresponde a la altura de x (por ejemplo 1.75m correspondería al número 175
Clasifique cada una de estas funciones.

5) Sean A, B dos conjuntos cualesquiera:
a) Determine A y B, para obtener una función de A en B que sea inyectiva.
b) Determine A y B, para obtener una función de A en B que sea sobreyectiva.
c) Determine A y B, para obtener una función de A en B que sea inyectiva y no sobreyectiva.
d) Determine A y B, para obtener una función de A en B que no sea inyectiva y sea sobreyectiva.
e) Determine A y B, para obtener una función de A en B que sea biyectiva.
f) Haga los diagramas de flechas para las funciones anteriores.

6) Sean A = {1,2,3} B = {a,b,c}
Sea f: A→B tal que f = {(1, a), (2, b), (3, c)}
Sea g: A→B tal que g = {(1, a), (2, a), (3, b)}
Sea h:A→B tal que h = { (1,a), (2,a), (3,a) }
a) Halle f(1), f(2), f(3), g(1), g(2), g(3), g(1), g(2), g(3)
b) Halle la función inversa f^{-1}. Halle $f^{-1}(a)$, $f^{-1}(b)$ y $f^{-1}(c)$

c) Halle los conjuntos preimagen $g^{-1}(a)$, $g^{-1}(b)$, $g^{-1}(c)$, $h^{-1}(c)$ y $h^{-1}(b)$

7) Sea el conjunto de los estudiantes de informática A = {pablo, ana, juan, lucía}
Sea el conjunto B = {1,…,12}
Tenemos una función ***f*** que relaciona a cada estudiante con su nota de aprobación

x	**f(x)**
pablo	8
ana	7
juan	7
Lucía	9

a) Escriba la función f de A en B
b) Obtenga el conjunto imagen de f
c) Clasifique la función f
d) Obtenga a partir de la función f los estudiantes que aprobaron con 7

8) Escriba la definición de función compuesta. Explique cuando dos funciones cualesquiera se pueden componer, o sea, que condición deben de cumplir. De un ejemplo de dos funciones que no se puedan componer y otro ejemplo donde sí se pueda.

9) Sean A = {1, 2, 3, 4} B = {x,y,z} C = {a,b,c,d} Sean f: A→B, g: B→C, definidas de la siguiente manera: f = {(1,x),(2,x),(3,z),(4,y)} g = {(x, a),(y,c),(z,d)}.
a) Halle la función compuesta g ° f.
b) Halle el conjunto Im (g ° f) (conjunto imagen de g ° f).
c) Escriba un diagrama de flechas que represente a la función compuesta de la parte anterior.

10) Sean A = {1, 2, 3, 4} B = {x, y, z} C = {a, b, c, d}
a) Escriba una función f de A en B, escriba una función g de B en C, luego halle g compuesta con f, o sea, halle la función g ° f.
b) Halle el conjunto Im (g ° f) (conjunto imagen de g ° f).
c) Escriba un diagrama de flechas que represente a la función compuesta de la parte anterior.

11) Sean A,B,C,D,E conjuntos no vacíos. Sean f: A→B, g: B→C, h: C→D, j: E→A
¿Cuál de las siguientes composiciones son posibles? Explique cuando sea posible cuál conjunto es el dominio y cuál el codominio en cada composición.
1. g ° f 2. g ° h 3. f ° j

12) Determine si las siguientes funciones son invertibles. En caso de que lo sean, halle la función inversa. Justifique, haga los correspondientes diagramas de flechas.
Sean A = {1, 2, 3, 4} B = {x, y, z} C = {a, b, c, d}
a) f: A→B tal que f = { (1,x), (2,x), (3,y), (4,y) }
b) g: A→C tal que i = { (1,a), (2,c), (3,b), (4,d) }
c) h: B→C tal que h = { (x,a), (y,b), (z,d)}
d) j: C→A tal que j = { (a,2), (b,1), (c,3), (d,4)}
e) i : A→C tal que f = { (1,a), (2,c), (3,b), (4,a) }

13) Para A = {1, 2, 3} y B = {3, 4,5}

a) ¿Cuántas funciones f de A en B satisfacen f(1) =4 y es Inyectiva? Explicar.
b) Encuentre una función f de A en B que no sea ni Inyectiva ni Sobreyectiva ni Biyectiva.

14) A={1,2,3,4} B={a,b,c} C={w,x,y,z}
Sean f:A→B y g:B→C tales que f = {(1,a),(2,a),(3,b),(4,c)} y g = {(a,x),(b,y),(c,z)}
Sean f:A→B y g:B→C tales que f = {(1,a),(2,a),(3,b),(4,c)} y g = {(a,x),(b,x),(c,x)}

a) Hallar g ° f en cada caso.
b) Clasifique g ° f en cada caso.

15) Sean f, g y h funciones definidas de la siguiente manera:
f: A → B :: **g:** B → C :: **h:** A → C tal que, h = {(a, x);(b, y);(c, z);(d, x)}

a) Determinar f y g sabiendo que g(1) = x ; g(2) = y ; g(3) = z y que A={a, b, c, d} B={1, 2, 3} ; C={x, y, z}
b) Clasificar f, g y h.
c) Determinar conjunto imagen de h.
d) Hallar la función inversa de g, (g^{-1}) [si es que esto es posible].
e) Demostrar que $g^{-1}(y) = 2g^{-1}(x)$.
f) Determinar el conjunto imagen de g^{-1}.

16) Sean f, g funciones definidas de la siguiente manera:
f: A → B
g: B → C con A={a, b, c, d} ; B={x, y, z, w} ; C={a, h, l, o, s}
h=f∘g
Se sabe además que f(a) = w, f(b) = x, f(c) = y.

a) Determinar f y g tal que h(b), h(c), h(d) y h(a) en este orden me permita "escribir" la palabra "hola".
b) Escribir la función h y clasificar f, g y h.
c) Hallar la función inversa de f, (f^{-1}) . Si esto es posible,
   i) determinar el conjunto imagen de f^{-1}.
   ii) Hallar $f^{-1}(y)$ y $f^{-1}(w)$.
d) Como modificaría g para que h(c), h(d) y h(a) [en este orden] me permita "escribir" la palabra "olas".

17) ****** Sea A={2,3,4,5,6,7,8} y B={1,2,3,4,7,11}
f: A→ B tal que a cada elemento **n** perteneciente a **A** se le asigna:
2n-3 si n es impar
n/2 si n es par.
Enumere los elementos de f e investigue si es inyectiva y sobreyectiva.

18) ******Sea **f** y **g** dos funciones definidas de **R** en **R** tal que f(x)= x+2 y g(x)= f(x)-x se pide:
Bosquejo de la Grafica de **f(x)** y **g(x).**
Demostrar que f(x) es inyectiva.
Sea h(x)= $2x^2$-3, componer **h** con **f(x).**
Determinar $f^{-1}(x)$. (sugerencia, usar $f^{-1}(x)=y$)

4 Grafos

4.1 Introducción

En matemáticas y ciencias de la computación, un grafo (del griego grafos: dibujo, imagen) o gráfica es el principal objeto de estudio de la teoría de grafos.
Informalmente, un grafo es un conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.
Típicamente, un grafo se representa gráficamente como un conjunto de puntos (vértices o nodos) unidos por líneas (aristas).
Desde un punto de vista práctico, los grafos permiten estudiar las interrelaciones entre unidades que interactúan unas con otras. Por ejemplo, una red de computadoras puede representarse y estudiarse mediante un grafo, en el cual los vértices representan terminales y las aristas representan conexiones (las cuales, a su vez, pueden ser cables o conexiones inalámbricas).

4.2 Terminología Básica

Un *grafo dirigido* **G** se define en términos abstractos como un par ordenado **(V,E)**, donde **V** es un conjunto y **E** es una relación binaria sobre **V**. Un grafo dirigido puede representarse geométricamente como un conjunto de puntos **V** con un conjunto de flechas **E** entre parejas de puntos. Por ejemplo, la figura 1 muestra un grafo dirigido. A los elementos de **V** los llamaremos vértices, y los pares ordenados de **E**, aristas del grafo dirigido. Se dice que una arista es incidente con los vértices que ella une. Por ejemplo, la arista (a,b) es incidente con los vértices **a** y **b**. En ocasiones, cuando deseemos ser más específicos, diremos que la arista (a,b) es incidente **a** desde y es incidente hacia **b**. El vértice **a** es llamado el vértice inicial y el vértice **b** el vértice Terminal de la arista (a,b). Una arista que es incidente a partir y hacia el mismo vértice, como (c,c) en la figura 1 es llamado un lazo. Se dice que dos vértices son adyacentes si están unidos por una arista. Además al referirnos a una arista (a,b). diremos que el vértice **a** es adyacente al vértice **b** y también diremos que el vértice **b** es adyacente desde el vértice **a**. También que un vértice es aislado si no hay una arista incidente con él.

Un *grafo no dirigido* **G** se define de manera abstracta como una par ordenado **(V,E)**, donde **V** es un conjunto y **E** es un conjunto de multiconjuntos de dos elementos de **V**. Por ejemplo, G=({a,b,c,d}, {{a,b,};{a,d};{b,c};{b,d};{c,c}}) es un grafo no dirigido. Un grafo no dirigido puede representarse geométricamente como un conjunto de puntos marcados **V** con un conjunto de líneas **E** entre los puntos. El grafo no dirigido **G** anterior se muestra en la figura 2. Veamos otro ejemplo: sea V={a,b ,c, d} un conjunto de cuatro jugadores en un torneo de tenis de eliminación directa. Sea E={(a,b);(a,d);(b,d);(c,a);(c,b);(dc)} una relación binaria sobre **V** de manera que (x,y) en **E** significa que x venció a y en el encuentro entre ellos. El grafo **G=(V,E)** se muestra en la figura 3. Entonces, de acuerdo con la figura 3, el jugador b (por ejemplo) es mejor que el jugador d quien, a su vez venció al jugador c.

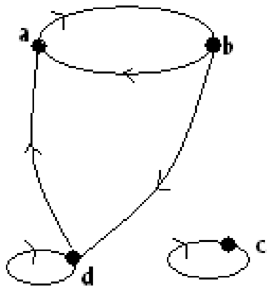


Figura 1

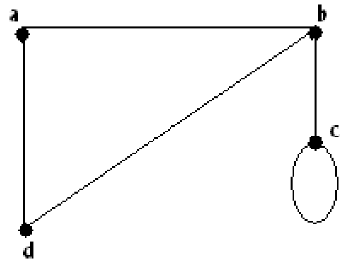


Figura 2

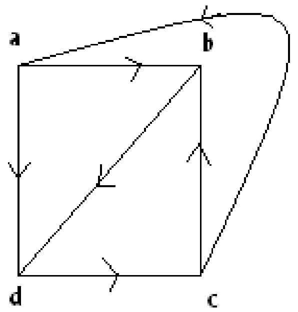


Figura 3

4.3 Multigrafos y Grafos Pesados

La definición de un grafo puede expresarse de varias maneras. Sea **G=(V,E)** donde **V** es un conjunto y **E** es un multiconjunto de pares ordenados de VxV. **G** es llamado un *multigrado dirigido*. Geométricamente, un multigrafo dirigido puede representarse como un conjunto de puntos marcados **V** con un conjunto de flechas **E** entre los puntos, donde no existe restricción en el número de flechas de un punto a otro punto (en efecto, la multiplicidad de un par ordenado de vértices en el multiconjunto **E** es el número de flechas entre los puntos marcados correspondientes). Por ejemplo, la figura 4 muestra un multigrafo. Ahora, consideremos una representación gráfica de un mapa de carreteras en la cual una arista entre dos ciudades corresponde a un carril de una autopista entre las ciudades. Como a menudo hay autopistas de varios carriles entre pares de ciudades, esta representación origina un multigrado. La noción de multigrado no dirigido puede definirse de manera similar. De ahora en adelante, cuando sea claro el contexto, usaremos el término grafo para significar ya sea un grafo o un multigrado, o ambos. Por otro lado, cuando sea necesario enfatizar que nos referimos a un grafo (en lugar de un multigrafo), usaremos el término grafo lineal.

Cuando modelamos una situación física con un grafo abstracto, hay muchas ocasiones en las cuales deseamos asociar información adicional a los vértices, a las aristas del grafo o a ambos. Por ejemplo, en un grafo que representa la conexión por autopista entre dos ciudades, podríamos desear un número a cada lado para indicar la distancia entre las dos ciudades conectadas por la arista. También podríamos desear asignar un número a cada vértice para indicar la población de la ciudad. En un grafo que represente los resultados de los encuentros del torneo de tenis podríamos etiquetar cada arista con las puntuaciones y la fecha del encuentro entre los equipos conectados por la arista. Los pesos pueden ser números, símbolos, o cualquier cantidad que deseemos asignar a los vértices y a las aristas.

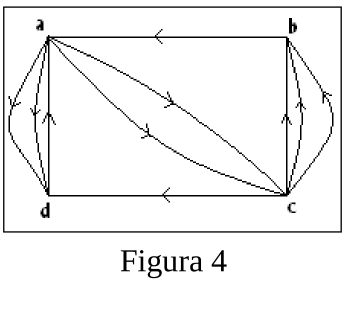


Figura 4

4.4 Definiciones

Camino.

Sean x e y vértices (no necesariamente distintos) de un grafo no dirigido G=(V,E). Un camino x-y en G es una sucesión alternada finita (sin lazos)

$$x = x_0, e1, x_1, e2\ x_2, e3, \ldots.., e_{n-1}, x_{n-1}, e_n, x_n = y$$

De vértices y aristas de G que comienza con el vértice **x** y termina con el vértice **y** y que contiene las n aristas $e_i = \{ x_{i-1}, x_i\}$ donde $1 \leq i \leq n$.

Longitud de un Camino.

La longitud de un camino es **n,** el número de aristas que hay en un camino. (Si n = 0, no existen aristas, x = y, y el camino se denomina trivial. Estos caminos no se tendrán muy en cuenta.)

Cualquier camino x-y donde x = y (y n>1) es un camino cerrado. En caso contrario, el camino es abierto.

Observe que un camino puede repetir aristas y vértices.

Recorridos, circuitos, caminos simple y ciclos.

Consideremos un camino x-y en un grafo no dirigido G = (V,E).

a) Si no se repite ninguna arista en el camino x-y, entonces el camino es un recorrido x-y. Un recorrido x-x cerrado es un circuito.
b) Cuando ningún vértice del camino x-y se presenta más de una vez, el camino es un camino simple x-y. El termino ciclo se usa para describir un camino simple cerrado x-x.

4.5 Resumen de las definiciones

Vértice(s) Repetido(s)	Aristas Repetida(s)	Abierto	Cerrado	Nombre
Si	Si	Si	----------	*Camino*
Si	Si	---------	Si	*Camino Cerrado*
Si	No	Si	----------	*Recorrido*
Si	No	---------	Si	*Circuito*
No	No	Si	----------	*Camino Simple*
No	No	---------	Si	*Ciclo*

4.6 Conexidad

Sea G = (V,E) un grafo no dirigido. Decimos que G es conexo si existe un camino simple entre cualesquiera dos vértices distintos de G.

Sea G = (V,E) un grafo dirigido. Su grafo no dirigido asociado e s un grafo obtenido de G si no se tienen en cuenta las direcciones de las aristas. Si se obtiene más de una arista no dirigida de un par de vértices distintos de G, entonces solo una de estas aristas se dibuja en el grafo no dirigido asociado. Cuando este grafo asociado es conexo, consideramos que G es conexo.

Un grafo que no es conexo es disconexo.

4.7 Grado de un vértice

Sea G un grafo o multigrado no dirigido. Para cualquier vértice v de G, el grado de **v**, que anotaremos grad(v), es el número de aristas en G que son incidentes con **v**.
En este caso, un lazo en un vértice **v** se considera como dos aristas incidentes en **v**.

4.8 Circuitos y Recorridos Eulerianos

Sea G = (V,E) un grafo o multigrafo no dirigido sin vértices aislados. Entonces G tiene un *Circuito Euleriano* si existe un circuito en G que recorre cada arista del grafo exactamente una vez. Si existe un recorrido abierto de **a** a **b** en G que recorre cada arista de G exactamente una vez, este recorrido se llamara *Recorrido Euleriano*.

4.9 Práctico de Grafos

1. Consideremos el problema de reconocer enunciados que en la lengua inglesa constan de un artículo, seguido de uno, dos o tres adjetivos, un sujeto y al último un verbo, como se muestra enseguida.

Ej:
El tren se detiene.
Una niña pequeña ríe.
Las nubes grandes, blancas, aborregadas, aparecen.

Cuando examinamos un enunciado palabra por palabra podemos determinar si tiene esta forma especial mediante el seguimiento de un grafo pesado que comenzara en un primer vértice al que nombraremos como **a**. Si el último vértice es alcanzado entonces el enunciado estará en la forma especial. Para simplificar el dibujo usaremos un único vértice para indicar el descubrimiento de palabras fuera del orden normal. En tal caso, alcanzando dicho vértice, significara la detección de un enunciado ilegal.

2. Para el grafo de la figura 5, determine:
   a) Un camino de **b** a **d** que no sea un recorrido
   b) Un recorrido **b-d** que no sea un camino simple.
   c) Un camino simple de **b** a **d**.
   d) Un camino cerrado de **b** a **d** que no sea un circuito.
   e) Un circuito de **b** a **b** que no sea un ciclo.
   f) Un ciclo de **b** a **b**.

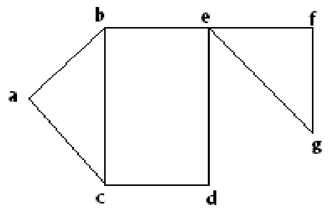
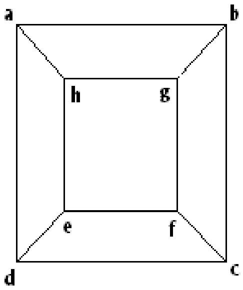


Figura 5

3. Para el grafo de la figura 5, ¿cuántos caminos simples existen de **b** a **f**?

4. Cuantos caminos simples diferentes existen entre los vértices a y f en el grafo dado en la figura 6.



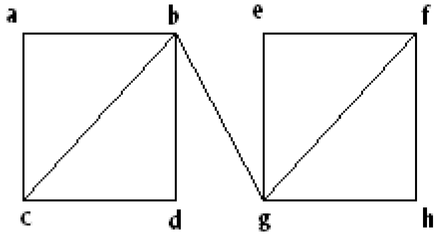


Figura 6 Figura 7

5. Sea G=(V,E) el grafo no dirigido de la figura 7. ¿Cuántos caminos simples existen en G de **a** a **h**? ¿Cuántos de ellos son de longitud 5?

6. Siete ciudades a, b, c, d, e, f y g están conectadas por un sistema de autopistas como sigue:

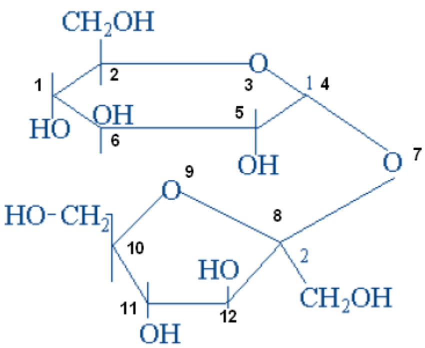
a) I-22 va de a a c, pasando por b.
b) I-33 va de c a d y entonces pasa por b y continúa hacia f.
c) I-44 va de d por e hacia a.
d) I-55 va de f a b pasando por g.
e) I-66 va de g a d.

Se pide:

(1) Use los vértices para las ciudades y las aristas dirigidas para los tramos de autopista que las unen, y dibuje un grafo dirigido que modele esta situación.
(2) Enumere los caminos simples de **g** a **a**.
(3) Cuál es el menor número de autopistas que tendría que cerrarse para interrumpir el paso de **b** a **d**.

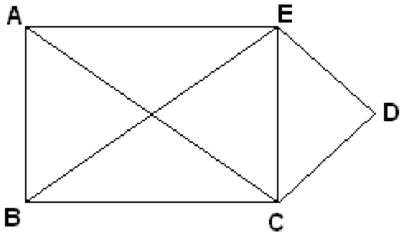
7. Sea G=(V,E) un grafo no dirigido. Definimos una relación R sobre V como aR b si a=b o si existe un camino simple en G de a a b. Demuestre que R es una relación de equivalencia.

8. Dada la siguiente molécula contestar lo que sigue:



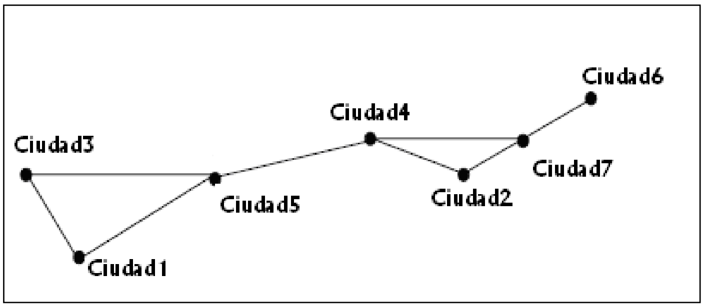
a) Modele la molécula con un grafo. Defina G=(V,E).
b) Defina camino simple. ¿Es posible ir del vértice diez (según figura) al vértice uno por un camino simple?
c) ¿Si quitáramos el átomo de oxígeno (numerado con siete) el grafo que representa la molécula se mantiene conexo? (Justifique).
d) Calcule la longitud de la molécula, asumiendo (para la resolución del ejercicio) que la longitud de todos los enlaces es similar.

9. Dado el siguiente grafo se pide:

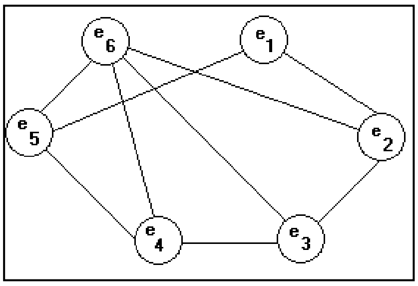


a) Defina G=(V,E)..
b) Determinar un camino simple de B a D.
c) Hallar la longitud de A a D (la menor).
d) ¿Si agregamos un vértice (F) a la figura, el grafo seguirá siendo conexo?

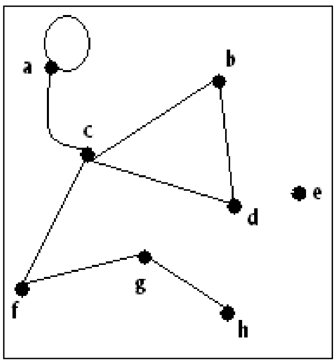
10. Se quiere duplicar los tramos de una red de comunicación que, en caso de deteriorarse, imposibilitarían la comunicación entre ciertos puntos.
a) ¿Cuáles son los tramos que deben duplicarse?
b) Determinar un camino simple desde la ciudad de Ciudad1a Ciudad7.
c) Determinar el o los circuitos simples del grafo de la figura.
d) Calcular la longitud del grafo.



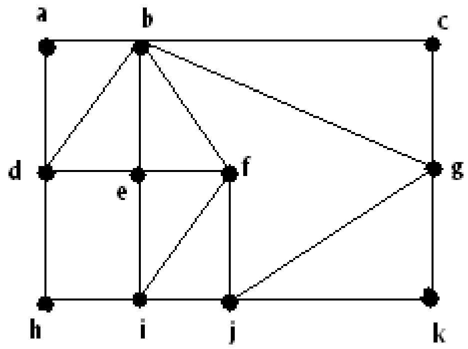
11. Definir el grafo G de la figura, (dar el conjunto de vértices y aristas).
a) Determinar un camino simple de e_2 a e_5
b) Dar un ejemplo de Circuito Simple y otro de Circuito Elemental.
c) Cuál es la distancia mínima de e_5 a e_2.



12. Dado el siguiente grafo G=(V,E) determinar el grado de cada uno de sus vértices. ¿Es G conexo? Explique.



13. Dado el siguiente grafo G=(V,E) determinar:
a) El grado de los vértices b, f y g.
b) Un circuito Euleriano.
c) La distancia del vértice **a** al vértice **k**.



14. Sea el conjunto A={1,2,3,4}. Se definen la siguiente relación:
R: B→B / R={(1,1),(1,3),(2,2),(2,4),(3,1), (3,2),(3,3),(3,4)}
a) Determinar el grafo dirigido de la relación R
b) Determinar todos los caminos simples desde el vértice 1 al 4

15. Se desea modelar el comportamiento de una máquina de ventas, la cual vende barras de chocolate por $15 la unidad. Por simplicidad, supongamos que la maquina acepta solo monedas de cinco y diez pesos y no regresa cambio cuando se han depositado más de $15. El grafo pesado es una descripción del comportamiento de la máquina, donde los vértices corresponden a las cantidades que ya han sido depositadas para la venta en un momento dado, a saber, 0,5, 10 y $15 o más. En cualquier momento, un cliente puede hacer una de tres cosas: depositar una moneda de $5, depositar una moneda de $10 y presionar un botón para un seleccionar una barra de chocolate en particular.

Se pide:
a) Modelar un grafo que represente el funcionamiento de la máquina.
b) Dar los conjuntos **V** de vértices y **E** de aristas del grafo **G** del problema.
c) ¿Como modificaría el grafo para que, en caso de que me pase de $15 la maquina me devuelva la moneda que agregue de más?
d) ¿Qué ocurriría si el precio de las golosinas aumenta a $20?,(teniendo en cuenta que se hacen los ajustes necesarios para que la maquina acepte billetes de veinte pesos).

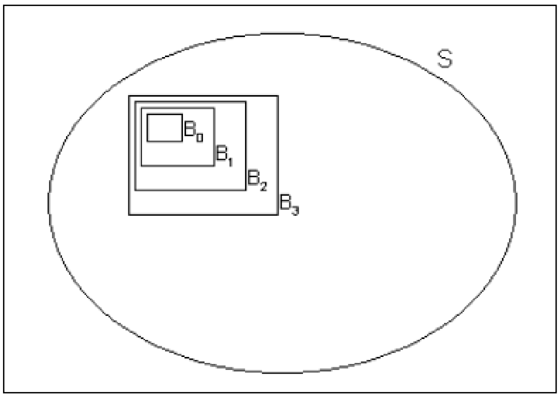
<u>**Nota:**</u> **(a las partes a) y b) del problema)**

En el grafo solución habrá tres aristas que salgan de cada vértice etiquetados con 5, 10 y P. Una arista con peso 5 actualiza la cantidad total depositada en la maquina cuando el cliente introduce una moneda de $5 y una arista con peso 10 actualiza la cantidad depositada en la maquina cuando el cliente introduce una moneda de $10.

5 Lenguajes

5.1 Esquema de inducción

Veremos la inducción como mecanismo primitivo para definir conjuntos. La idea es tomar un conjunto conocido *S*, partir de algún subconjunto $B_0 \subseteq S$, definir formas de agrandar sucesivamente B_0 y considerar el conjunto que se obtiene al límite de este proceso.
Intuitivamente....
La idea de la definición inductiva de un conjunto es, a partir de ciertos elementos individuales, agregar nuevos elementos en el conjunto combinando los ya existentes.
Definir inductivamente un conjunto significa dar las reglas que indican cómo construir los elementos del conjunto.



El conjunto *S* debe ser un conjunto conocido y podrá ser por ejemplo el conjunto *N* de los números naturales o el conjunto de las tiras de caracteres sobre determinado alfabeto.

Llamamos Lenguaje a un conjunto de palabras (tiras, secuencias, strings, ...) construidas sobre un conjunto dado de símbolos. Ese conjunto de símbolos se le llama Alfabeto del lenguaje. Hay lenguajes que se pueden definir inductivamente y tratar como conjuntos inductivos.

En general ...
• La idea de la Definición Inductiva de Conjuntos es:
– Agregar ciertos elementos individuales en el conjunto.
– Construir nuevos elementos del conjunto combinando los elementos agregados anteriormente.
• Las definiciones suelen escribirse como reglas que debe cumplir el conjunto que se está definiendo.

Ejemplo:
¿Cómo definir los Naturales?
– Dando una regla que diga que 0 es un natural.
– Dando otra regla que diga que, si tenemos un natural n, se puede construir otro natural aplicando el operador sucesor (o sumando 1).
• Definición Inductiva de los naturales.
– 0 $\in$ ***N.***
– Si n $\in$ ***N, entonces n+1*** $\in$ ***N.***

5.2 Definición Inductiva de un Conjunto

Una definición inductiva de un conjunto ***A*** *consiste en una colección de esquemas de reglas. Cada esquema es de tipo:*
Básicos: *si establecen que ciertos elementos pertenecen al conjunto.*
Inductivos: *si establecen que un elemento está en el conjunto si ciertos otros elementos están en el conjunto.*
Clausura: *si establece que los únicos elementos del conjunto, son los que se pueden construir aplicando los demás esquemas dados un numero finito de veces.*

Ejemplo Numérico
• Naturales:
– 0 $\in N$
– $n+1 \in N$, *si* $n \in N$
• Pares
– 0 $\in P$
– $n + 2 \in P$, *si* $n \in P$
• Impares
– 1 $\in I$
– $n+2 \in I$, *si* $n \in I$

Observar que:
• En las reglas inductivas, las metavariables (***n*** en este caso) siempre representan elementos que necesitan menos reglas para su construcción.

5.3 Significado de una definición inductiva

• Cuando damos una definición inductiva de un conjunto:
1. Definimos un conjunto.
2. Definimos una manera de recorrer(construir) sus elementos.
3. Todos los elementos del conjunto se recorren con las reglas dadas.
4. <u>El orden al aplicar las reglas es relevante, dado que un orden distinto puede dar (y en general lo hace) elementos distintos.</u>

A considerar…
En una definición inductiva puede haber más de una cláusula base y/o más de una inductiva., puede no haber cláusulas inductivas, pero lo que obligatoriamente tiene que estar definido es <u>al menos una cláusula base.</u>
La cláusula de clausura se verá muy similar en todas las definiciones inductivas, por lo tanto, muchas veces la podemos omitir o simplemente escribir Cláusula de Clausura (CC).

5.4 Lenguajes

• Llamamos **Lenguaje** a un conjunto de frases o palabras (tiras, secuencias, strings, ...) construidas sobre un conjunto dado de símbolos.
• Ese conjunto de símbolos se le llama **Alfabeto** del lenguaje.
• Hay lenguajes que se pueden definir inductivamente y tratar como conjuntos Inductivos.

Ejemplo: {a,b}*
• Sea {a,b} un conjunto de símbolos (alfabeto)
• {a,b}* es el conjunto de todas las posibles secuencias formadas con los símbolos a y b
• {a,b}* se define inductivamente por las siguientes cláusulas:
i. e $\in$ *{a,b}* (palabra vacía)*
ii. Si w $\in$ *{a,b}* entonces aw* $\in$ *{a,b}**
iii. Si w $\in$ *{a,b}* entonces bw* $\in$ *{a,b}**
• Observar que ***e*** es una palabra y no un símbolo del alfabeto!

Más ejemplos….

- L1 inc ex. {a, b}
  i. a $\in$ L1
  ii. Si w $\in$ L1 entonces bwb $\in$ L1

- L2 inc. ex. {a, b, c}
  i. b $\in$ L2
  ii. Si w $\in$ L2 entonces awc $\in$ L2

5.5 Pertenencia a un conjunto Inductivo

Para probar que un objeto pertenece a un conjunto inductivo, basta con mostrar como lo formamos (su secuencia de formación está dada por las cláusulas utilizadas).
Ejemplo: bbabb $\in$ L1 pues: a $\in$ L1, bab $\in$ L1, bbabb $\in$ L1
Otros ejemplos… veamos otra definición inductiva del conjunto de los números pares:

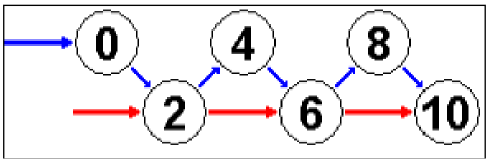
Tomemos:
• Como *S* a los naturales
• Como base al conjunto {0}
• Como única regla: *"si n es par, entonces n* + 2 *es par"*

Una observación interesante en este punto es que podríamos haber realizado otra definición inductiva para el mismo conjunto de los números pares:

Tomemos:
• Como *S* a los naturales
• Como base al conjunto {0,2}

• Como única regla: *"si n es par, entonces n + 4 es par"*

Los conjuntos son iguales, pero la forma de recorrer los elementos es distinta. Así, demostrar que 10 es par podría implicar seguir distintos caminos (e incurrir en distintos costos), dependiendo de la definición inductiva que se considere.



Otro enfoque para las mismas ideas…
Sea S un conjunto cualquiera.
S^* = {w| w * = es una palabra con letras del alfabeto S}
S^+ = {w| w es una palabra no vacía con letras del alfabeto S}

- A la palabra vacía la seguimos anotaremos ε o e (como se vio anteriormente).
- A un conjunto de palabras se lo denomina lenguaje. S^* y S^+ son ejemplos de lenguajes para cualquier alfabeto S, pero nos concentraremos en lenguajes definidos de forma recursiva.
- Consideremos además que:
  Al conjunto de todas las palabras posibles se lo conoce también como **Σ*** y a **Σ** como alfabeto

Por ejemplo:

Sea $S = \{a,b,c\}$, y $L_1 \subseteq S^*$ definido inductivamente por:
i. a $\in L$ (cláusula base: no depende de otros objetos de L_1)
ii. Si $w \in L_1$, entonces $bwb \in L_1$ (cláusula recursiva: depende de otros objetos de L_1)

Sea $S = \{a,b,c\}$, y $L_2 \subseteq S^*$ definido inductivamente por:
i. 2 $b \in L$
ii. Si 2 $w \in L_2$, entonces 2 $awc \in L_2$

Con una definición inductiva no sólo definimos un conjunto, sino que decimos la forma de construir objetos en él, e implícitamente decimos que es la única forma de construir objetos en el conjunto.
Para probar que un objeto pertenece a un conjunto inductivo basta con mostrar cómo lo formamos (su secuencia de formación está dada por las cláusulas utilizadas).
Por ejemplo:

1 $bbabb \in L_1$ ya que:
i. $a \in L_1$ (aplicamos i)
ii. Luego, 1 $bab \in L_1$ (aplicamos ii)
iii. Finalmente, 1 $bbabb \in L_1$ (aplicamos ii)

En el ejemplo anterior, se hace evidente que todas las palabras del lenguaje L_1 tendrán un número par de símbolos b, pero ¿qué es "tener un número par de símbolos b "?

Más ejemplos…

El conjunto $\sum^*$ es el mínimo conjunto **X** que satisface:

i) $\varepsilon \in \mathbf{X}$
ii) Si $\boldsymbol{\alpha} \in \mathbf{X}$ entonces $\mathbf{a}\boldsymbol{\alpha} \in \mathbf{X}$
iii) Si $\boldsymbol{\alpha} \in \mathbf{X}$ entonces $\mathbf{b}\boldsymbol{\alpha} \in \mathbf{X}$

En este caso la clausura está dada al principio al decir que **X** es el mínimo conjunto que satisface.

Preguntas: 1) ¿Cuál sería $\sum$? ¡defínalo!
2) ¿Que piensa Ud. del conjunto **X**? ¿Cómo lo interpreta Ud.?

5.6 Práctico de Lenguajes

16. Sea {a, b, c} alfabeto y L lenguaje en el alfabeto definido por las siguientes reglas:

a) $a \in L$

b) Si $x \in L$, $xc \in L$

c) Si $x \in L$, $xb \in L$.

¿Cuáles de las siguientes afirmaciones son verdaderas o falsas, justifique?

i) $bb \in L$

ii) $abb \in L$

iii) $abc \in L$

iv) $abcc \in L$

v) Si $x \in L$, entonces largo de x es impar.

vi) Si $x \in L$, x comienza con a.

17. Sea B un lenguaje definido sobre V = {0,1} inductivamente donde:

$1 \in B$

$0 \in B$

Si $\alpha \in B \rightarrow 1\alpha 1 \in B$

Si $\alpha \in B \rightarrow 0\alpha 0 \in B$

C. Clausura

Y el siguiente criptograma donde existen cadenas de ese lenguaje de largo mayor de 6 escondidas encuéntrelas

101100111010111110010101010
1000010000101000110101010101
0101010001000111110111110101
10000011100000001101010101000
1000101010001111100010011010

Una de ellas es de largo quince y es la clave de acceso a un sistema ¿Cuáles? Justifique su respuesta.

18. Considere el conjunto U = R. Dé una definición alternativa (por comprensión, por extensión o si es un conjunto conocido dar su nombre) de los siguientes conjuntos definidos inductivamente:

a) Sea A el conjunto definido inductivamente por las cláusulas:

i. $0 \in A$

ii. Si $n \in A$ entonces $(n+3) \in A$.

iii. Estos son todos los elementos del conjunto.

b) Sea B el conjunto definido inductivamente por las cláusulas:

i) $8 \in B$

ii) Si $n \in C$ entonces $(n+4) \in B$,

iii) Si $n \in C$ entonces $(n-4) \in B$.

iv) Estos son todos los elementos del conjunto.

19. a) Definir inductivamente el conjunto de los naturales múltiplos de 6 (o sea $\{0, 6, 12, \ldots\}$).
b) Definir inductivamente el conjunto de los enteros múltiplos de 7.

5) a) Define inductivamente el lenguaje Σ^* sobre el alfabeto $\Sigma = \{0, 1\}$.
b) Define inductivamente el lenguaje Σ^* sobre el alfabeto $\Sigma = \{a, b, c\}$.

6) Define inductivamente los siguientes lenguajes sobre el alfabeto $\Sigma = \{a, b, c\}$:
a) El lenguaje {ε, c, c c, c c c, c c c c,...}
b) El lenguaje {ε, ab, aabb, aaabbb,...}
c) El lenguaje {bc, bcbc, bcbcbc, ...}
d) El lenguaje de las palabras que comienzan con la letra b.
e) El lenguaje de las palabras que terminan con la letra a.
f) El lenguaje de las palabras que son palíndromos.

7) Sea $\Sigma = \{a, b\}$. Sea $A_1 \subseteq \Sigma^*$ definido inductivamente por las cláusulas:
i. $\varepsilon \in A_1$
ii. Si $\alpha \in A_1$ entonces $\alpha bb \in A_1$
iii. Si $\alpha \in A_1$ entonces $\alpha a \in A_1$
iv. Estos son todos los elementos del conjunto.

8) ¿Cuáles de las siguientes afirmaciones son correctas? Justifique su respuesta en términos de la aplicación de las cláusulas.
1) $a \in A_1$ 2) $b \in A1$ 3) $bbabb \in A_1$ 4) $abba \in A_1$ 5) $bbaabb \in A_1$
6) $bbbabbb \in A_1$ 5) $aaaaa \in A_1$

9) Sea $\Sigma = \{a,b\}$. Sea $A_2 \subseteq \Sigma^{**}$ definido inductivamente por las cláusulas:
i. $a \in A_2$
ii. Si $\alpha \in A_2$ entonces $b\alpha b \in A_2$
iii. Si $\alpha \in A_2$ entonces $\alpha a \in A_2$
iv. Estos son todos los elementos del conjunto.

¿Cuáles de las siguientes afirmaciones son correctas? Justifique su respuesta en términos de la aplicación de las cláusulas.
1) $b \in A2$ 2) $bab \in A2$ 3) $ba \in A2$ 4) $babab \in A2$ 5) $aba \in A2$ 6) $bbabb \in A2$

10) Sea $\Sigma = \{a, b, c\}$. Sea $A3 \subseteq \Sigma^{**}$ definido inductivamente por las cláusulas:
i. $\varepsilon \in A3$
ii. Si $\alpha \in A_3$ entonces $b\alpha c \in A_3$
iii. Si $\alpha \in A_3$ entonces $b\alpha a \in A_3$
iv. Estos son todos los elementos del conjunto.

Escriba 5 palabras que pertenezcan a A3 y 3 que no pertenezcan.

6 Recursividad

6.1 Definiciones recursivas

Comenzamos en esta sección analizando la sucesión de enteros b_0, b_1, b_2, b_3, ..., donde b_n= 2n para todo n $\in$ N. Tenemos que b_0=2.0=0, b_1=2.1=2 b_2=2.2=4 y b_3=2.3=6. Si, por ejemplo, necesitamos determinar b6, simplemente calculamos b_6=2.6=12, sin necesidad de calcular el valor de b_n para cualquier otro n $\in$ N. Podemos realizar estos cálculos ya que tenemos una formula explicita, b_n=2n, que nos dice como determinar b_n conociendo n (solamente).
Examinemos ahora la sucesión de enteros a_0, a_1, a_2, a_3,...., donde:

a_0=1, a_1=2, a_3=3, y
$a_n=a_{n-1}+a_{n-2}+a_{n-3}$ para todo n $\in$ Z^+ tal que $n\geq 3$.

Aquí no tenemos una formula explicita que defina cada en términos de n, para todo n n $\in$ N. Si queremos conocer el valor de a6, por ejemplo, necesitamos los valores de a_5, a_4 y a_3. Y estos valores (los de a_5, a_4 y a_3) requieren que conozcamos también los valores de a_2, a_1 y a_0. A diferencia de la situación más fácil en que determinamos b_6=2.6=12, para calcular a_6 tendríamos que escribir:

$a_3=a_2+a_1+a_0=3+2+1=6$
$a_4=a_3+a_2+a_1=6+3+2=11$
$a_5=a_4+a_3+a_2=11+6+3=20$
$a_6=a_5+a_4+a_3=20+11+6=37$

Sin importar como llegamos a a_6, vemos que las dos sucesiones de enteros (b_0, b_1, b_2, b_3, ..., y a_0, a_1, a_2, a_3, ...) no solo son numéricamente diferentes. Los enteros b_0, b_1, b_2, b_3, ..., se pueden enumerar fácilmente como 0, 2, 4, 6, ... y para cualquier n $\in$ N tenemos la formula explicita b_n=2n. Por otro lado, tal vez sea difícil (si no es que imposible) determinar tal formula explicita para los enteros a_0, a_1, a_2, a_3, ...
A veces es difícil definir un concepto matemático de manera explicita. Pero como en el caso de la sucesión a_0, a_1, a_2, a_3, ..., podríamos definir lo que necesitamos en términos de otros resultados anteriores similares. Cuando hacemos esto, decimos que el concepto está definido *en forma recursiva*, usando el método o proceso de *recursión*. de esta manera obtenemos el concepto que nos interesaba estudiar, por medio de una *definición recursiva*. Por lo tanto, aunque no tengamos una formula explicita en el caso de la sucesión a_0, a_1, a_2, a_3, ..., si tenemos una forma de definir los enteros a_n para n $\in$ N, por recursión. Las asignaciones a0=1, a1=2, a2=3 proporcionan una base para la recursión.

La ecuación

$a_n=a_{n-1}+a_{n-2}+a_{n-3}$ para todo n $\in$ Z^+ tal que $n\geq 3$,

proporciona el proceso recursivo; indica la forma de obtener nuevos elementos de la sucesión a partir de los resultados anteriores ya conocidos (o que se pueden calcular).

Un ejemplo de conjunto definido de forma recurrente es el de los números naturales:

a) 0 pertenece a **N**
b) Si *n* pertenece a **N**, entonces *n*+1 pertenece a **N**
c) Si **X** verifica a) y b) , entonces **N** está incluido en **X**

6.2 Funciones definidas de forma recurrente

Aquellas funciones cuyo dominio puede ser recursivamente definido pueden ser definidas de forma recurrente.

¡El ejemplo más conocido es la definición recurrente de la función factorial n!:

$$n! = \begin{cases} \text{si } n = 0 & \Rightarrow 1 \\ \text{si } n \geq 1 & \Rightarrow n\,(n-1)! \end{cases}$$

Con esta definición veamos cómo funciona esta función para el valor del factorial de 3:

3! = 3 · (3-1)!
= 3 · 2!
= 3 · 2 · (2-1)!
= 3 · 2 · 1!
= 3 · 2 · 1 · (1-1)!
= 3 · 2 · 1 · 0!
= 3 · 2 · 1 · 1
= 6

El ejemplo del cálculo recursivo del factorial de un número llevado al campo de la programación, en este ejemplo C++:

```
int factorial(int x)
{
  if (x > -1 && x < 2) return 1;  // Cuando -1 < x < 2 devolvemos 1 puesto que 0! = 1 y 1! = 1
  else if (x < 0) return 0;       // Error no existe factorial de números negativos
  return x * factorial(x - 1);    // Si x >= 2 devolvemos el producto de x por el factorial de x - 1
}
```

El seguimiento de la recursividad programada es casi exactamente igual al ejemplo antes dado, para intentar ayudar a que se entienda mejor se ha acompañado con muchas explicaciones y con colores que diferencia los distintos sub-procesos de la recursividad.

```
X = 3 //Queremos 3!, por lo tanto X inicial es 3
X >= 2 -> return 3*factorial(2);
  X = 2 //Ahora estamos solicitando el factorial de 2
```

X >= 2 -> return 2*factorial(1);
X = 1 // Ahora estamos solicitando el factorial de 1
X < 2 -> return 1;
[En este punto tenemos el factorial de 1 por lo que volvemos marcha atrás resolviendo todos los resultados]
return 2 [es decir: return 2*1 = return 2*factorial(1)]
return 6 [es decir: return 3*2 = return 3*factorial(2)*factorial(1)] // El resultado devuelto es 6

6.3 Definición de f Recursiva

Sea **A** un conjunto inductivo. Una función **f: A → B**, se define mediante ecuaciones que determinan:

- El **valor de f** para los objetos de A obtenidos de **aplicar las cláusulas base.**
- El **valor de f** para los objetos de A obtenidos de **aplicar las cláusulas inductivas**, utilizando el valor de f en los objetos anteriores y también el valor de los objetos anteriores.

Ejemplos…

1) Defina una función que calcula la cantidad de símbolos de una palabra de Σ^* siendo $\Sigma=\{a, b\}$. Entonces **f: Σ^* → N**
Seguimos esquema de la definición inductiva del conjunto.

1. $f(\varepsilon) = 0$
2. $f(a\alpha) = 1 + f(\alpha)$
3. $f(b\alpha) = 1 + f(\alpha)$

Para calcular la cantidad de símbolos de la palabra bba, la aplicación de la función nos queda de la siguiente forma:

$f(bba) = 1+f(ba)$ (por cláusula 3 de la definición de f)
$f(ba) = 1+f(a)$ (por cláusula 3 de la definición de f)
$f(a) = 1+f(\varepsilon)$ (por cláusula 2 de la definición de f)
$f(\varepsilon) = 0$

Luego se sustituyen los valores «sucesivamente, hacia atras» obteniendo el valor buscado, en este caso $f(bba) = 3$

2) Escribir una función que aplicada a una palabra de Σ^* siendo $\Sigma=\{a, b\}$, devuelve otra palabra obtenida de intercambiar las a y las b.

Solución
$f: \Sigma^* \rightarrow \Sigma^*$ tal que

1. $f(\varepsilon) = \varepsilon$
2. $f(a\alpha) = bf(\alpha)$
3. $f(b\alpha) = af(\alpha)$

6.4 Práctico Funciones Recursivas

(1) Considera el lenguaje Σ^* siendo $\Sigma = \{a, b , c\}$. Se define la siguiente función recursiva f: $\Sigma^* \rightarrow N$ tal que:

i. $f(\varepsilon)= 4$
ii. $f(a\alpha) = f(\alpha)+1$
iii. $f(b\alpha) = f(\alpha)$
iv. $f(c\alpha) = f(\alpha)+3$

Calcular: f(a), f(b), f(c), f(ac), f(bca) y f(bcca)

(2) Sea $\sum = \{a,b\}$. Sea $A_2 \subseteq \sum^*$ definido inductivamente por las cláusulas:

i. $a \in A_2$
ii. Si $\alpha \in A_2$ entonces $b\alpha b \in A_2$
iii. Si $\alpha \in A_2$ entonces $\alpha a \in A_2$
iv. Estos son todos los elementos del conjunto.

Sea $z \in A_2$ defina:

a) f: $A_2 \longrightarrow N$ /f(z) devuelve la cantidad de **a** de z.
b) g: $A_2 \longrightarrow N$ /g(z) devuelve la cantidad de **b** de z.
c) j: $A_2 \longrightarrow N$ /j(z) devuelve una palabra que solo tiene las **a** de **z**
e) k: $A_2 \longrightarrow N$ /k(z) intercambia las **b** con las **a** de **z**

Hallar: f(babaa), g(baba), j(baba) y k(babaa).

(3) Sea $\Sigma = \{a,b,c\}$. Sea $A_3 \subseteq \Sigma^*$ definido inductivamente por:

i. $\varepsilon \in A_3$
ii. Si $\alpha \in A_3$ entonces $b\alpha c \in A_3$
iii. Si $\alpha \in A_3$ entonces $b\alpha a \in A_3$
iv. Estos son todos los elementos del conjunto.

I) Sea $z \in A_3$, Defina:

a) f: $A_3 \longrightarrow N$ /f(z) cuenta la cantidad de **b** de z. Hallar f(bbca).
b) g: $A_3 \longrightarrow N$ /g(z) suprime las b de z. Hallar g(bbca).

II) Investiga comportamiento de h: $A_3 \longrightarrow \Sigma^*$ /

1) $h(\varepsilon)= \varepsilon$
2) $h(b\alpha c) = h(\alpha)bc$;
3) $h(b\alpha a)= h(\alpha)ba$

Hallar h(bbbaca)

(4) Sea Σ = {a,b,c }. Sea $A_7 \subseteq \Sigma^*$ definido inductivamente por:

i. $\varepsilon \in A_7$

ii. Si $\alpha \in A_7$ entonces $b\alpha bc \in A_7$

iii. Si $\alpha \in A_7$ entonces $b\alpha ba \in A_7$

iv. Estos son todos los elementos del conjunto.

Sea $z \in A_7$, Defina:

a) f: $A_7 \rightarrow$ N /f(z) devuelve la cantidad de **a** de z.

b) g: $A_7 \rightarrow$ N /g(z) devuelve la cantidad de **b** de z.

c) h: $A_7 \rightarrow$ N /h(z) devuelve la cantidad de **c** de **z**

d) j: $A_7 \rightarrow$ N /j(z) cuenta la cantidad de **c** y **a** de **z**

e) m: $A_7 \rightarrow \Sigma^*$ /m(z) cambia todos los símbolos por **a**.

Determine cada valor funcional para z=bbbabc

7 Haskell

7.1 Cuarenta Años de Diseño y Construcción

Haskell se fundó en la ciudad de Jacksonville, Florida en 1965, comprometida con el concepto de generar proyectos de diseño y construcción completamente integrados. A finales de la década de los sesenta, Haskell expandió sus actividades de proyectos multifamiliares a construcción de industrias y oficinas, además de agregar servicios de desarrollo incluyendo adquisición de inmuebles y financiamiento.

A medida que el crecimiento de la empresa continuó durante los años setenta, muchas de las empresas más importantes de Estados Unidos comenzaron a utilizar la presentación de diseño y construcción para sus instalaciones y consideraron a Haskell como una fuente única.

La reputación de Haskell como líder en el diseño y la construcción continuó en la década de los ochenta cuando las autoridades públicas adoptaron este método de elaboración de proyectos. Se agregaron los servicios de Manejo de Construcción y se estableció la Comunidad de Desarrolladores Haskell para proporcionar servicios al mercado constituido por la comunidad para el cuidado de personas retiradas.

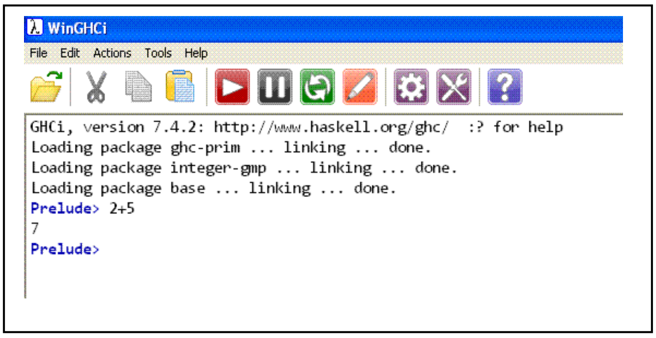
Durante la década de los noventa, Haskell refinó su estructura para enfocarse en segmentos específicos del mercado, para lo cual estableció oficinas en Dallas, Texas y la Ciudad de México. En 1993 Haskell jugó un papel decisivo en la formación del Instituto de Diseño y Construcción de América (DBIA, por sus siglas en inglés).

Apoyada en bases sólidas y con una reputación construida durante más de cuarenta años, Haskell sigue siendo líder en el diseño y la construcción a través del aumento de sus beneficios y la capacitación de sus practicantes.

Instalación de Haskell.

Descargue el instalador desde: www.haskell.org una vez instalado pruebe algunas operaciones simples de suma, resta, multiplicación, etc., en este punto vera que Haskell se comporta como una calculadora sencilla.

Se trabajará con la versión para Windows **WinGHCi** que se encuentra como un archivo ejecutable en la propia instalación. Si lo ejecutamos se verá así:



Funciones predefinidas.
Observe las siguientes funciones predefinidas y de ser posible pruebe su funcionamiento en Haskell.

div a b : el cociente de la división entera entre a y b.

El cociente de la división entera entre 33 y 7 es 4. Aquí se llama **div.**

div 33 7 = 4, observe que no lleva paréntesis. Sólo un espacio después de la palabra div y otro espacio entre el 33 y el 7.

La función **div** espera 2 números enteros y su respuesta es otro número entero.

¿Siempre se puede hacer la división entre a y b, para cualquier a y b enteros?

$$\text{div}: Z \times Z^{*} \rightarrow Z$$

Esto significa que la función **div** tiene por dominio un par ordenado de números enteros, en el cual el primer elemento es un número entero, cualquiera, y el segundo elemento es un número entero diferente de cero. Además, el condominio es un número entero. Es importante entender que significa todo esto; resulta esencial contrastar el formalismo matemático con su correcta interpretación y posterior aplicación a ejercicios prácticos.

Algunos errores en los que se incurre habitualmente; si digita *div 33 4 6* el programa dará un mensaje de error. Hay demasiados parámetros en este caso.
Si ud. digita *div 33* el programa le dará otro mensaje de error, ya que en este caso hay pocos parámetros para realizar la operación.

Revise lo siguiente…

mod a b : resto de dividir a

entre b mod 33 7 = 5

mod 7 33 = 7

Ejercicio: ¿Que espera obtener si digitamos mod 19 3? ¿Y si digitamos mod (-19) 3?
(Observación: los números negativos necesitan paréntesis en Haskell)
Recuerde que en la definición de división entera es fundamental que el resto de la división entera siempre sea menor que el divisor y además positivo.

División entera entre a y b

Definición: Sean a y b números enteros, con b positivo. Dividir a entre b significa encontrar dos números enteros q y r de modo que se verifique:

a=b*q + r, con 0 <= r < b

Vamos a dividir (-19) entre 3. ¿Cuál de las siguientes opciones será la correcta?

-19	3	-19	3	-19	3
-4	-5	-1	-6	2	-7

En los tres casos se cumple que (-19) = 3.q + r, pero sólo en uno el resto es positivo y menorque 3.

¿Quieres probarlo con tu computadora? Recuerda escribir los números negativos entre paréntesis. mod (-19) 3   div (-19) 3

Funciones lógicas: | | (or)   && (and)   == (igual)

Las funciones lógicas devuelven un elemento del conjunto **Bool.** El conjunto **Bool** es unconjunto que tiene solamente 2 elementos: Verdadero y Falso.

Boole = {True, False} = {Verdadero, Falso} = {0,1}

- Nos preguntamos si 8 será mayor que 5. La respuesta es sí, por supuesto.En haskell se escribe así: 8>5. Al dar enter la espuesta es True.
- (8 > 5) && (3 > 7) En este caso la primera expresión es verdadera pero la segunda no. Entonces la respuesta será False.
- También nos podremos preguntar si el resto de dividir 9 entre 2 será 0. **mod 9 2 == 0** (Observe que hay 2 signos de = pegados. No se puede dejar espacio entre ellos). La respuesta es False.

En resumen:

mod 9 2 tiene como respuesta 1 (Calcula el resto de la división entera de 9 entre 2)

mod 9 2 ==1 tiene respuesta True. (pregunta ¿El resto de la división entera de 9 entre 2 es 1?)

mod 9 2 ==0 tiene respuesta False. (pregunta ¿El resto de la división entera de 9 entre 2 es 0?)

Otras funciones lógicas predefinidas en Hugs son por ejemplo: /= (distinto), not (negación).

Definición de funciones en Haskell:

En Haskell se pueden definir nuevas funciones.

Para definir una función primero hay que crear un archivo de texto nuevo, con extensión .hs

Ejemplo 1: Empecemos con la función

f : Z →Z / f(x) = 2x

En una carpeta, creamos un archivo prueba.hs (puede ser el mismo archivo que usamos antes)

Lo abrimos con Haskell y escribimos en una línea, el dominio y codominio.

En otra línea escribimos la fórmula de la función. La notación es la siguiente:

```
f :: Integer -> Integer
f x = 2 * x
```

En la primera línea estamos indicando que la función se llamará "f" y que su dominio es el conjunto de los enteros, llamados Integer en Haskell, y el codominio es también Integer.

Los cuatro puntos forman parte de la notación. La flecha, que separa el dominio del codominio, se hace con el signo de menos y el signo mayor juntos y sin espacio.

No dejar espacio en el medio, porque dará error.

->correcto

- >error

La separación entre la letra f y los cuatro puntos, así como los demás espacios se han puesto sólo para facilitar la lectura. El programa no les adjudica valor a los espacios de más.

f(x)=2x se escribe en Haskell f x = 2*x. Ahora, habiendo escrito esto en el archivo prueba.hs, lo guardamos, cerramos y recargamos. Si ahora, en la pantalla de comandos, escribimos f 8 y luego enter, debería aparecer el 16.

Ejemplo 2:

Se pueden definir funciones con más de una variable. Por ejemplo, recordemos que una fórmula muy utilizada en física es que F=m.a (Fuerza = masa * aceleración).

Podemos también definirla con Haskell. Pero el inconveniente es que en el mismo archivo no pueden coexistir dos funciones con el mismo nombre. En nuestro archivo prueba.hs ya tenemos una función llamada "f". Hacemos un archivo nuevo o podemos llamarla de otra forma.

Por ejemplo, podemos llamarle a la fuerza, fu. Entonces, el archivo prueba.hs, editado, quedará así:

```
x=45
fu :: Integer -> Integer
fu a b = a*b
```

La función fu tiene como dominio un entero y otro entero, y su codominio es otro entero. Con la notación matemática habitual, escribiríamos fu (4,5)=20

En haskell, escribimos fu 4 5 = 20

Definición de funciones en Haskell: El conjunto Bool.

El conjunto de los números enteros tiene infinitos elementos y es el que usamos hasta ahora. El conjunto Bool tiene sólo 2 elementos: True y False (verdadero y falso).

Ejemplo: queremos implementar una función contraseña que servirá para evaluar si un número ingresado por el operador es el correcto en una contraseña. Por ejemplo, supongamos que la contraseña es 3214. La idea entonces es que nuestra función recibe un número entero y la respuesta es correcto o incorrecto. Dicho de otra forma, la respuesta es True o False.

password :: Integer ->Bool

password 3214 = True

password a = False

(¡Implemente esta función en Haskell y pruebela!)

Funciones recursivas en Haskell

Una función recursiva es una función que se llama a sí misma en algún paso previo.

Observación: Tiene que existir alguna condición de terminación, también llamada condición de parada.

Observación 2: si no es función, no puede ser función recursiva.

Ejemplo:

tio::Integer->Integer

tio n = 9 + tio (n-1)

Intentaremos evaluar ahora tio 5

Aplicando esta función, vemos que, si sustituimos n por 5 queda:

tio 5 = 9 + tio (5-1) O sea, haciendo la resta, tio 5 = 9 + tio 4

Ahora volvemos a aplicar la función para calcular tio 4 y ya hacemos las operaciones: tio 4 = 9 + tio 3. Y aplicando de nuevo la función…

tio 3 = 9 + tio 2

tio 2 = 9 + tio 1

tio 1 = 9 + tio 0

tio 0 = 3 + tio (-1)

tio (-1) = 3 + tio (-2)

Y así seguimos en forma infinita........ Aparecerá un mensaje de error. Falta la condición de terminación. Esta "función" está, entonces, mal definida. “No es una función”. Vamos a arreglarla.

```
tio::Integer->Integer
tio 2 = 100
tio n = 9 + tio (n-1)
```

Entonces, ahora que sabemos que tio 2 = 100 podemos calcular, de abajo hacia arriba: tio 3 = 9 + tio 2 y como tio 2 = 100 entonces tio 3 = 109

tio 4 = 9 + tio 3. y como tio 3 = 109 entonces tio 4 = 118

tio 5 = 9 + tio 4 y como tio 4 = 118 entonces tio 5 = 127 Es la respuesta final. Lo que hicimos anteriormente se llama secuencia de cómputos.

En resumen, con esta función se puede calcular tio 5, tio 994, tio 3000.

¿Se podrá calcular tio 1 ? ¿Y se podrá calcular tio (-4)?

Esta función, tio, ¿es una función total o parcial, en los enteros?

Expresiones condicionales: *la sentencia condicional if.*

La cláusula if significa "si", es un si condicional.

Primero vamos a escribirlo en español:

si (.......) entonces (...................) y sino entonces ( )

si (llueve) entonces (vamos al cine) y sino entonces (vamos al tablado)

si (la entrada cuesta menos de $200) entonces (vamos al cine) y sino entonces (vamos a caminar y te invito un helado)

En resumen:

si (condición lógica) entonces (acción 1) y sino entonces (acción 2)

Dentro del primer paréntesis no se puede escribir sólo un número, una palabra o una frase. Tiene que ser un valor booleano. Es una condición lógica, por tanto sus valores son True o False.

Ejemplo:

Implementaremos la función edad. Si el número de entrada es menor que 18, le suma 3 unidades. Si es mayor o igual que 18, le resta 2 unidades. Entonces, si todo marcha bien, debería quedar así:

edad (14) = 17

edad (17) = 20

edad (18) = 16

edad (25) = 23

Una forma de definir dicha función es:

```
edad:: Integer-> Integer
edad a = if ( a<18 ) then ( a+3 ) else ( a-2 )
```

Si el valor de a es menor de 18, el paréntesis (a<18) tiene el valor True, entonces lo que se ejecuta es (a+3). Cuando a no es menor de 18, el paréntesis (a<18) tiene el valor False, entonces lo que se ejecuta es (a-2).

El signo de igual = ==

Asignación: Cuando escribimos j=3, estamos diciendo que el valor de j es 3. Antes j no tenía ningún valor previo, y a partir de ahora sabemos que el valor de j es 3. Le estamos asignando un valor a la variable j: el valor 3. Más aún. Aunque j antes tuviera otro valor, a partir de la expresión j=3 el valor de j será 3.

Comparación: Otra situación diferente es comparar el valor de una variable con otro. La variable m tiene un valor, el cual no conocemos. Quizás sea 17. Quisiéramos comparar el valor de m, que no sabemos cuál es, con el número 17.¿Será m igual a 17 o no? Cuando nos hacemos una pregunta lógica, el signo de igual es ==.Esto es, son 2 signos de igual juntos, sin espacio entre ellos.

Si m valía 17, la expresión m==17 tiene valor True.

Si m no valía 17, la expresión m==17 tiene valor False.

Sea cual sea el valor de m la expresión m==17 no cambia el valor original de m. No le estamos asignando ningún valor a m. Sólo se lo está comparando.

Ahora estamos en condiciones de poder hacer la función h.

$$h: Z+ \rightarrow Z+ \ / \quad h(x)= \begin{cases} 10 \text{ si } x \text{ es par} \\ 11 \text{ si } x \text{ es impar} \end{cases}$$

Lo nuevo que vamos a ver ahora es que se puede usar en haskell una función como insumo, como una parte integrante de la misma. En esta función h tenemos que discriminar según el valor de entrada, según sea par o no.. Pero como ya tenemos hecha la función par, la podemos utilizar.

Para poder utilizar varias funciones simultáneamente, una forma de hacerlo es escribirla en el mismo archivo. En nuestro caso, todo está escrito en el archivo prueba.hs

Ya hicimos la función par. Ahora llegó el momento de utilizarla.

Si la entrada de la función h es un número que lo llamamos x, el resultado dependerá de saber si x es par o impar.

Recordemos que si x es par, entonces par x tiene el valor True.

```
h:: Integer-> Integer
h x = if  (par x) then ( 10 ) else ( 11 )
```

Entonces, si x es par, (par x) tiene valor True y la respuesta es 10. Si x es impar, (par x) tiene valor False y la respuesta es 11.

Dada la siguiente función decidir qué cosa hace:

par::Integer->Bool
par 0 = True
par 1 = False
par n = par (n-2)

Para comprobar o testear si la función par hace lo que quisiéramos, vamos a efectuarla secuencia de cómputos de algún número, por ejemplo, el 6
par 6 = par 4
par 4 = par 2
par 2 = par 0
par 0 = True ¡¡Entonces sí, el 6 es par !!
¿La pregunta que se puede hacer en este punto es que ocurrirá si hacemos par 7?

Listas

Conjuntos definidos por inducción: Listas. Funciones sobre Listas.

Ahora introduciremos una estructura matemática denominada listas o secuencias, con la cual se pueden modelar muchas realidades. Una lista es una ordenación de elementos de un cierto conjunto, una lista puede ser vacía.

Ejemplos de listas diferentes, con elementos del conjunto {3, 4, 7} 1) [3, 4, 7, 7]

2) [3, 4, 7, 7, 7]

3) [4, 3, 3, 3]

4) [3, 4, 7, 7, 3]

5) [3, 4, 7]

6) [3, 3, 3, 3, 3, 3, 3]

En las listas es importante el orden de los elementos que la integran. Además, los elementos repetidos también importan.

NOTACIÓN: Para indicar listas, lo haremos señalando entre paréntesis rectos sus elementos. Por ejemplo, la lista que contiene el 1, el 1 y el 5 será [1, 1, 5]. Esta lista es diferente de la siguiente: [5, 1, 1]

La lista vacía, esto es, la lista que no tiene elementos, la simbolizaremos [ ].

Mas ejemplos: con los elementos del conjunto A = {3}, ¿cuántas listas se pueden formar?

Respuesta:

1) [ ]

2) [3]

3) [3, 3]

4) [3, 3, 3]

5) [3, 3, 3, 3]

Si el conjunto A no es el conjunto vacío las listas que se pueden formar tomando elementos de A son infinitas. Cada lista tiene una cantidad finita de elementos, pero la cantidad de listas que se pueden formar con los elementos de un conjunto dado A, es un número infinito.

Sea A un conjunto cualquiera. Definiremos el conjunto de todas las listas que se pueden construir con elementos de A por inducción. Llamaremos a este conjunto ***list A***.

Definición inductiva del conjunto ***list A***

1) La lista vacía pertenece a list A

2) Si tenemos un elemento de A y una lista que pertenece a list A, podemos formar una nueva lista agregándole al principio de dicha lista ese nuevo elemento. Por ejemplo, si tenemos la lista [5,7,2] y el elemento a agregar es el 4, entonces la nueva lista será 4:[5,7,2] = [4,5,7,2]

3) Las únicas maneras de formar listas es aplicar las reglas 1 y/o 2 un número finito de veces.

Veamos ahora la misma definición, pero con la notación usada en Haskell.

1. Cláusula base: [ ] pertenece a ***list A***
2. Clausula inductiva: Si b pertenece a A y x pertenece a ***list A*** entonces b:x pertenece a ***list A***
3. Cláusula de clausura: CC

La cláusula o regla 1 dice que la lista vacía es una lista.

La cláusula o regla 2 dice que si **b** es un elemento de **A** y **x** es una lista, le podemos agregar a la lista **x** un nuevo elemento, el **b**, al comienzo, formándose una nueva lista, **b:x**, que tiene un elemento más que **x**.

La cláusula o regla 3 es la cláusula de clausura, y lo que dice es que solamente podemos formas listas aplicando las cláusulas o reglas 1 y 2 un número finito de veces.

Ejemplo: Supongamos que tenemos el conjunto A = {1, 2, 7}

Por aplicación de la regla número 1, la lista vacía pertenece a **list A.** Como el número 2 pertenece al conjunto A y la lista vacía pertenece a **list A**, entonces aplicando la segunda regla, se le puede agregar a la lista vacía el 2. Queda entonces 2: [ ] = [2]

Como el número 1 pertenece al conjunto A y la lista [2] pertenece a **list A**, entonces aplicando la segunda regla, se le puede agregar a dicha lista el 1. Queda 1: [2] = [1,2]

Como el número 7 pertenece al conjunto A y la lista [1,2] pertenece a **list A**, entonces aplicando la segunda regla, se le puede agregar a dicha lista el 7. Queda 7: [1,2] = [7,1,2]

Como el número 1 pertenece a A y la lista [7,1,2] pertenece a **list A**, entonces aplicando la segunda regla, se le puede agregar a dicha lista el 1. Queda 1: [7,1,2] = [1,7,1,2]

Como el número 1 pertenece a A y la lista vacía pertenece a **list A**, entonces aplicando la segunda regla, se le puede agregar a dicha lista el 1. Queda 1: [ ] = [1]

Y así seguimos construyendo más elementos de list A.

Notación de Haskell:

- ✓ [a] es una lista con exactamente un elemento
- ✓ [a, b] es una lista con exactamente 2 elementos
- ✓ [a, b, c] es una lista con exactamente 3 elementos
- ✓ (a) es una lista con cualquier cantidad de elementos; podría ser también la lista vacía.
- ✓ (a:x) es una lista donde **a** es el primer elemento de la lista y **x** el resto de la lista; tienepor lo menos un elemento, el a.
- ✓ (a:b:x) es una lista donde **a** es el primer elemento de la lista, **b** es el segundo y x el resto de la lista; tiene por lo menos 2 elementos.

Ejemplo: si escribimos en el editor de textos:

(a:x) referido a la lista [7,2,4,1] significa que a es 7 y que x es el resto, esto es, [2,4,1].

(a:x) referido a "manzana" significa que a es la 'm' y que "anzana" es x.

Funciones definidas por recursión sobre listas

Ejemplo 1: Queremos definir una función, llamada **largo**, que nos proporcione el largo de una lista de números enteros. Por ejemplo, si digitamos **largo [6, 2, 3, 1]** esperamos obtener un 4, porque esta lista tiene 4 elementos.

largo [True, True, False] = 3; **largo "cuatro"** = 6, **largo [6,2,3,1]**= 4

Observación: largo [6, 2, 3,1] = 1 + largo [2,3,1]

Generalizando**: largo (a:x) = 1 + largo (x)**, donde x es una lista cualquiera.

Ya podemos *"intentar"* hacer la función **largo**. (Esta función, como todas las funciones, para implementarla en Haskell, hay que escribirla en el **editor de texto**).

largo :: [Integer]- Integer

```
largo :: [Integer] -> Integer
largo (a:x) = 1+ largo (x)
```

Antes de probarla con Haskell intentaremos probar la función en el papel. A esto se le llama escribir la "secuencia de cómputos".
Secuencia de cómputos para largo [8,5,0,9] largo [8,5,0,9] = 1 + largo [5,0,9]
largo [5,0,9]= 1 + largo [0,9]
largo [0,9] = 1 + largo [9]
largo [9] = 1 + largo [ ]
largo [ ] = … No lo podemos completar, ya que no hemos indicado en ningún lado cuanto debería valer el largo de la lista vacía. Si lo probamos en Haskell nos dará un mensaje de error.

¡¡Falta definir el valor de la función largo cuando aplicamos la lista vacía!!

```
largo :: [Integer] -> Integer
largo [] = 0
largo (a:x) = 1+ largo (x)
```

¡Ahora sí, estaría definida correctamente!

Practico de Haskell

1. Escriba en la línea de comandos, de a una en una las siguientes expresiones en Haskell, y cada vez pulse la tecla ENTER.

a) 20*33

b) 2^18

c) 24*(67-111)

d) 2^(6+3)

e) 73/5

f) div 73 5

g) mod 73 5

h) mod (-20) 3

i) div (-20) 3

j) Deduzca el comportamiento de las funciones div y mod, predefinidas en Hasskell

k) Pruebe sus propios ejemplos

2. Escriba y ejecute las siguientes expresiones en Haskell.

a) max 7 9

b) min 2 (-8)

3. En Haskell no se pueden realizar definiciones en línea de comandos, para ello se deberá utilizar un archivo.hs. Pruebe ejecutar la siguiente línea en Haskell:

y = x+1

Cree un archivo en la carpeta Mis Documentos de su disco y denomínelo primero.hs

Copie las siguientes líneas en su archivo y guarde.

y = x+1

x = 2*3

Ejecutar en Haskell File/open/ y luego busque primero.hs y selecciónelo.

Escriba x en la línea de comandos y dé enter, lo mismo con y.

Idem con

y = x^2

x = (2*3)+(4-5)

4. Calcular en Haskell:

a) (8>2) || (4>1)

b) (8>2) && (4>1)

c) (8>2) && (4<1)

d) Escribir y probar en Haskell sus propias expresiones utilizando los operadores lógicos introducidos.

5. Escriba y evalúe las siguientes expresiones en Haskell.

a) 7<0

b) 8>2

c) div 8 4 == 2

d) div 4 3 == 0

e) mod 8 4 == 0

f) not (mod 8 4 == 7)

g) 4 /=4

h) div 8 9 /= 20

i) mod 50 10 <= 0

j) div 50 10 >= 6

k) Agregue sus propias expresiones booleanas en Haskell y evalúelas

6. Realizar los siguientes cálculos en Haskell:

a) Calcular 2^2

b) Calcular 2^4

c) Calcular 2^32

d) Calcular 2^64

e) Calcular 2^200

f) Calcular 2^4::Int

g) Calcular 2^32::Int

h) Calcular 2^32::Integer

i) Calcular 2^31::Int

j) Calcular 2^31 – 1::Int

k) Calcular min Bound::Int y max Bound::Int y comparar con los resultados de 9 y 10

l) Estudiar los resultados obtenidos y analizar posibles explicaciones

7. Verificar en Haskell:

a) :type 'a'

b) :type "abcdf"

c) :type True

d) :type 4<5

e) :type "a"

f) length "perro"

g) Deducir el tipo de length

h) Verificar :type length

i) Escribir y probar en Haskell sus propias expresiones

8. Probar en Haskell:

a) (4, 'a', "gato")

b) :type (4, 'a', "gato")

c) :type ('a', "zapato")

d) :type ("haskell","matematica")

e) :type (True, 5)

f) :type ('?'," pepito ")

g) :type ("cualquiera",(3.14,False))

9. Probar en Haskell:

a) fst ("jota", (1,'b'))

b) snd ("jota", (1,'b'))

c) Indicar el tipo de fst y snd y verificarlo en Haskell.

10. Abrir en el bloc de notas el archivo primero.hs

Copiar al archivo primero.hs las siguientes definiciones de funciones:

```
f :: Int -> Int
f n = 2^n

g :: Int -> Integer
g n = 2^n
```

Estudiar su significado.

a) Calcular f 2

b) Calcular g 4

c) Calcular f 32

d) Calcular f 64

e) Calcular f 200

f) Calcular g 200

g) Calcular g 35

i) Explicar los resultados.

11. Comenzando con funciones

1. Definir una función maximo::(Integer, Integer)->Integer que devuelve el mayor de sus dos argumentos.

2. Definir una función par::Integer->Bool que indica si su argumento es par (Sugerencia: Utilizar el operador ‘mod‘).

3. Definir una función max3 ::(Integer, Integer, Integer)->Integer que devuelve el máximo de sus argumentos.

4. Definir la función signo:: Int->Int que dado un número devuelve 1, 0 ó -1, en caso que el número sea positivo, cero o negativo respectivamente.

5. Definir la función abso::Int->Int que calcula el valor absoluto de un número.

6. Definir el predicado bisiesto::Int->Bool que determina si un año es bisiesto. Los años bisiestos son aquellos que son divisibles por 4 pero no por 100 a menos que también lo sean por 400. Por ejemplo, 1900 no es bisiesto pero 2000 sí lo es.

7. Tres números positivos pueden ser la medida de los lados de un triángulo si y sólo si el mayor de ellos es menor que la suma de los otros dos. Definir una función lados_triangulo::(Float, Float, Float)-> Bool que devuelva True si los tres números que se le pasan verifican esta condición, y False en caso contrario.

8. Definir una función es_rectangulo::(Integer, Integer, Integer)->Bool que devuelva True si los números que se le pasan pueden ser los lados de un triángulo rectángulo, y False en caso contrario. Sugerencia: una manera sería ordenar los tres números y verificar si el cuadrado del mayor de ellos es igual a la suma de los cuadrados de los otros dos. Sin embargo existe otra manera más fácil, utilizando (sólo una vez) la función max_3 ::(Integer, Integer, Integer)- >Integer que devuelva el máximo de tres enteros. ¿Se te ocurre? Probar la función con las entradas (3,5,4), (5,13,12) y (7,3,5).

9. ¡¡¡Definir una función para calcular el área de un círculo, dado su radio r. Usar “pi” que Haskell lo entiende !!!

Soluciones a alguno a los ejercicios de funciones:

```
dobleDe :: Int->Int
dobleDe x=2*x
```

funcion de la diapo 16

```
dobleDos :: Int -> Int-> Int
dobleDos x y = x*2 + y*2

dobleDosbis :: Int -> Int-> Int
dobleDosbis x y = dobleDe x + dobleDe y

anterior :: Int-> Int
anterior x = x-1
```

ejemplo diapo 3 de funciones-2

```
f::Int->Int
f x =if (x<10) then 2*x else x-1

signo::Int->Int
signo n = if n<0 then -1 else if n == 0 then 0 else 1
```

Ejemplos de concordancia de patrones

```
lucky :: Integer -> String
lucky 7 = "El siete de la suerte"
lucky x = "Lo siento no es tu dia de suerte"

esTres :: Integer -> Bool
esTres 3 = True
esTres x = False

esA::Char->Bool
esA 'a' = True
esA 'A' = True
esA x   = False
```

EJERCICIOS DE LA PRESENTACIÓN

Definir una función que devuelve el mayor de sus dos argumentos.

```
maximo :: (Int, Int) -> Int
maximo (x,y) = if x>y then x else y
```

Definir una función que indica si su argumento es par

```
par :: Int -> Bool
par x = if (mod x 2 ==0 ) then True else False
```

Definir una función que devuelve el máximo de sus argumentos.

```
max3 :: (Int, Int, Int) -> Int
```

```
max3 (x,y,z) = if (x>y && x>z) then x else if(y>x && y>z) then y else z
```

Definir la función que calcula el valor absoluto de un número.

```
abso :: Int -> Int
abso x = if x>=0 then x else (-x)
```

Definir el predicado que determina si un año es bisiesto.
Los años bisiestos son aquellos que son divisibles por 4 pero no por 100 a menos que también lo sean por 400.
Por ejemplo, 1900 no es bisiesto pero 2000 sí lo es.

```
bisiesto :: Int -> Bool
bisiesto x= if ((mod x 4==0) && (mod x 100 /=0))||(mod x 400==0) then True else False
```

Uso de ecuaciones con guardas

```
min3 :: Int->Int->Int->Int
min3 x y z | x <= y && x <= z  = x
           | y <= x && y <= z  = y
           | otherwise = z

abso2 :: Int -> Int
abso2 x |  x>=0 =x
        | otherwise =(-x)
```

E8.del práctico 5: Definir una función es rectángulo::(Integer, Integer, Integer)->Bool que devuelva True si los números que se le pasan pueden ser los lados de un triángulo rectángulo, y False en caso contrario. Sugerencia: una manera sería ordenar los tres números y verificar si el cuadrado del mayor de ellos es igual a la suma de los cuadrados de los otros dos. Sin embargo existe otra manera más fácil, utilizando (sólo una vez) la función max_3 ::(Integer, Integer, Integer)-- >Integer que devuelva el máximo de tres enteros. ¿Se te ocurre? Probar la función con las entradas (3,5,4), (5,13,12) y (7,3,5).

```
es_rectangulo::(Int, Int, Int)->Bool
es_rectangulo (x,y,z)= if (max3 (x,y,z))^2==x^2+y^2+z^2-(max3 (x,y,z))^2 then True else False
```

otra forma, usando guardas…

```
es_rect :: (Int, Int,Int)->Bool
es_rect (x,y,z) | x^2== y^2+z^2  =True
                | y^2==x^2+z^2   =True
                | z^2==y^2+x^2   =True
                | otherwise      =False
```

Definir una función para sumar los elementos de una lista de números enteros.

```
sumaLista::[Int]->Int
sumaLista [] = 0
sumaLista (x:xs)= x +sumaLista xs
```

Dada una lista de enteros, "extraer" de ella los números menores que 10, obteniendo otra lista con dichos números.

```
*Main> menor10 [3,16,5,2,7,25,45]
[3,5,2,7]

menor10 :: [Int]->[Int]
menor10 []= []
menor10 (x:xs) = if x<10 then x:(menor10 xs) else menor10 xs
```

sumar los menores que 10

```
sumamenor10 :: [Int]->Int
sumamenor10 []= 0
sumamenor10 (x:xs) = if x<10 then x+(sumamenor10 xs) else sumamenor10 xs
```

12. Dada una lista de enteros, devolver la lista ordenada de menor a mayor

```
borrar::[Int]->Int->[Int]   --RECUERDEN QUE ESTE EJERCICIO EXCEDE LO QUE
TRABAJAMOS EN EL CURSO
borrar [] x =[]
borrar (y:p) x = if (x==y) then (borrar p x) else  [y] ++ (borrar p x)

ordenada::[Int] -> [Int]
ordenada [] = []
ordenada [x]=[x]
ordenada (x:y:[])= if x <= y then [x,y] else [y,x]
ordenada (x:p) = if x <= head (ordenada p) then x:(ordenada p)
          else (head (ordenada p)):ordenada (x:(borrar (ordenada p) (head (ordenada p))))
```

Ejercicio 8

```
sacarLugar::Int->[Int]->[Int]
sacarLugar n []=[]
sacarLugar n (x:s)= if (n==1) then s else x:(sacarLugar (n-1) (s))

sacarLugar 2 [10,5,7,4]= 10:(sacarLugar 1 [5,7,4])
sacarLugar 1 [5,7,4]= [7,4]
```

con el resultado obtenido, volvemos a armar la lista (linea 130)

```
 sacarLugar 2 [10,5,7,4]= 10:([7,4])=[10,7,4]
```

multi4no8 que recibe un numero entero y devuelve True si es múltiplo de 4 y no de 8

```
multi4no8::Int->Bool
multi4no8 x = if mod x 4==0 && mod x 8/=0 then True else False
```

Examen que recibe una lista de notas y devuelve otra solo con las notas mayores que 7

```
examen ::[Int]->[Int]
examen []=[]
examen (x:s) = if x<=7 then examen (s) else x:(examen s)
```

Ver comportamiento de la siguiente función

```
incognita ::[Int]->Int
```

incognita [] = 0
incognita (x:s) = if (x<30) then x+(incognita s) else incognita s
incognita [10,22,40,36]=10+(incognita [22,40,36]

Dada una lista, devolver el promedio de sus elementos
promL ::[Int]->Int
promL []= error "lista vacia"
promL (xs)= div (sumaLista (xs)) (length (xs))

Investiga el resultado de aplicar el siguiente codigo a la lista [1..15]
ejcuatro::[Int]->[Int]
ejcuatro []=[]
ejcuatro (x:p)=if (mod x 3 ==0) then x:(ejcuatro p) else ejcuatro p

Ejercicios de repaso parte B b
func::[Int]->[Int]->[Int]
func [] [] =[]
func [] (p)= (p)
func (p) []= (p)
func (x:p) (y:s)= func (p) (s)

multi5 que recibe 2 números enteros, devuelve True si
ambos números son múltiplos de 5, en cualquier otro caso devuelve falso.

multi5 :: Int->Int->Bool
multi5 x y = if mod x 5 == 0 && mod y 5 == 0 then True else False

4. diferenciacifras que recibe un número entero positivo
menor que 100 y devuelve la diferencia de sus cifras (siempre positivo)

dc::Int->Int
dc x = abs (div x 10 - mod x 10)

```
dci ::Int->Int
dci x | x>99 = error "ingrese  número de dos cifras"
      | x<10 = error "ingrese  número de dos cifras"
      | otherwise = abs (div x 10 - mod x 10)
```

5. Sumacifra que recibe un número entero positivo menor que 100 y devuelve True
siempre que la suma de sus cifras sea par.

```
sc::Int->Bool
sc x | x>99 = error " ingrese  número de dos cifras"
     | x<0 = error " ingrese  número de dos cifras"
     | mod (div x 10+ mod x 10) 2==0 = True
     |otherwise = False
```

7. Tiene8 que recibe un número entero positivo de hasta 3 cifras y devuelve True
si alguna de sus cifras es 8, en caso contrario devuelve False.-}
t8 :: Int->Bool

```
t8 x | x>999 = error " ingrese  numero de dos cifras"
     | x<0 = error " ingrese  numero de dos cifras"
             | mod x 10 == 8 = True
             | div x 10 == 8 = True
             | div x 100 == 8 = True
             | mod (div x 10) 10 == 8 =True
             | otherwise = False
```

otra forma…

```
ti8 :: Int->Bool
ti8 x = if ( (mod x 10 == 8) || (div x 10 == 8) || (div x 100 == 8) || (mod (div x 10) 10 == 8))  then
True else False
```

8.Ultima igual recibe dos números y devuelve true si ambos números terminan con la misma cifra.

```
ultimaigual::Int->Int->Bool
ultimaigual x y | x == y =True
                | mod x 10 == mod y 10  = True
                | otherwise =False
```

otra forma…

```
ui::Int->Int->Bool
ui x y = if mod x 10 == mod y 10  then True else False
```

9.Func toma un num entero, devuelve dicho numero por 10 si este es menor que 100, divido 10 si es mayor que mil, en otro caso el mismo número.

```
fiun :: Int->Int
fiun x | x <100  = x*10
       | x > 1000 = div x 10
       | otherwise = x
```

Analice que hace la siguiente función.

```
multi4no82 :: Int->Bool
multi4no82 x = if (mod x 4 == 0) && (mod x 8 /= 0) then True else False
```

Algunos ejemplos mas de funciones implementadas en Haskell…

```
--Función recursiva para calcular el factorial de un número
   factorial :: Integer -> Integer
   factorial n = if n==0 then
                       1
                 else
                       n * factorial (n - 1)


   --Función recursiva para calcular el factorial de un número usando
pseudónimos
   factorial :: Integer -> Integer
   factorial 0         = 1
   factorial m@(n + 1) = m * factorial n

   --Función para calcular las raíces de una ecuación de segundo grado a partir
de sus coeficientes
   raíces :: Float -> Float -> Float -> (Float, Float)
   raíces a b c
    | disc >= 0 = ((-b + raizDisc) / denom,
                   (-b - raizDisc) / denom)
    | otherwise = error "La ecuación tiene raíces complejas"
      where
         disc = b*b - 4*a*c
         raizDisc = sqrt disc
         denom = 2*a
```

8 Bibliografía

Material extraído de:

- **Ralph P. Grimaldi** – *Matemáticas Discreta y Combinatoria.* Tercera edición.

- **C. L. Liu** – *Elementos de Matemáticas Discretas*. Segunda Edición. Departamento de Ciencias de la Computación. University of Illinois at Urbana-Champaign - M^c Graw Hill

- **Ross / Wright** – *Matemáticas Discretas*. Department of Mathematics, University of Oregon - Segunda Edición. Prentice Hall

- **Kenneth H. Rosen**. *Matemática discreta y sus aplicaciones*.

- **Echenique, Patricia/ Echenique, Paula/ Saul Tenenbaum** - Matemática I (2017). *Profesorado de Ciencia de la Computación.*

- **Profesora Rosana Álvarez** - Profesorado de Matemáticas, INET – (Inducción y Recursividad, Haskell) :: (http://x.edu.uy/)

- http://**www.fing.edu.uy**/tecnoinf/cursos/mdl1/material/teo/logica/log.teorico1.pdf
- http://www.fing.edu.uy/inco/cursos/logica/teorico/2012/02_12_Induccion.pdf
- http://www.haskell.com/es/Company/Profile/History/
- http://www.cs.us.es/~jalonso/cursos/i1m/ejercicios/ej_prog_Haskell.pdf
- http://www.x.edu.uy/inet/ppt%20haskell.pdf
- http://www.fceia.unr.edu.ar/lcc/r213/archivos/12.LFyC.Conjuntos.pdf