

Resumen Matemática Discreta

Jonathan David Rosenblatt

Primer Cuatrimestre, 2019

Índice

1. Trabajo Práctico I	3
1.1. Lógica y Conjuntos	3
1.2. Álgebras de Boole	4
1.3. Isomorfismos entre Álgebras de Boole	5
1.4. Relaciones de Orden	6
1.5. Relaciones de Equivalencia	7
1.6. Autómatas	7
2. Trabajo Práctico II	8
2.1. Inducción Matemática	8
2.2. Relaciones de Recurrencia	8
3. Trabajo Práctico III	10
3.1. Introducción a la Teoría de Grafos	10
3.2. Grafos Ponderados	15
3.3. Grafos Orientados	16

1. Trabajo Práctico I

1.1. Lógica y Conjuntos

p	q	p'	$p \cdot q$	$p + q$	$p \oplus q$	$p \Rightarrow q$	$p \Leftrightarrow q$	$p \downarrow q$	$p \uparrow q$
0	0	1	0	0	0	1	1	1	1
0	1	1	0	1	1	1	0	0	1
1	0	0	0	1	1	0	0	0	1
1	1	0	1	1	0	1	1	0	0

Tabla de valores lógicos.

Recordemos que todas las funciones lógicas tienen su contraparte en funciones de conjuntos. Particularmente $pq \equiv P \cap Q$, $p + q \equiv P \cup Q$, $\mathbf{F} \equiv \emptyset$, $\mathbf{T} \equiv \mathbf{U}$, $p \rightarrow q \equiv P \subset Q$, $P - Q \equiv pq'$.

Propiedades varias :

Conmutatividad : $p + q = q + p$, $pq = qp$

Distributividad : $p(q + r) = pq + pr$, $p + qr = (p + q)(p + r)$

Neutros : $p + \mathbf{F} = \mathbf{F}$, $p\mathbf{T} = \mathbf{T}$

Complemento : $p + p' = \mathbf{T}$, $pp' = \mathbf{F}$, $\mathbf{F}' = \mathbf{T}$, $\mathbf{T}' = \mathbf{F}$

Acotación : $p + \mathbf{T} = \mathbf{T}$, $p\mathbf{F} = \mathbf{F}$

Asociativa : $(p + q) + r = p + (q + r)$, $(pq)r = p(qr)$

Involución : $(p')' = p$

DeMorgan : $(p + q)' = p'q'$, $(pq)' = p' + q'$

Idempotencia : $p + p = p$, $pp = p$

Absorción : $p + pq = p$, $p(p + q) = p$

Nor : $(p + q)' = p'q' = p \downarrow q$, $p = (p \downarrow p)'$

Nand : $(pq)' = p' + q' = p \uparrow q$, $p = (p \uparrow p)'$

$x \leq y \Leftrightarrow xy = x \Leftrightarrow xy' = 0 \Leftrightarrow x' + y = 1$

$X = Y \Leftrightarrow X \subseteq Y \wedge Y \subseteq X$

$x = y \Leftrightarrow x'y + xy' = \mathbf{F}$

$X \cup Y = \emptyset \Rightarrow X = \emptyset \wedge Y = \emptyset$

$X \cap Y' = \emptyset \equiv X \subset Y \equiv Y' \subset X'$

Forma conjuntiva: producto de sumas.

Forma disyuntiva: suma de productos.

Forma normalizada: aparecen todas las variables en todos los términos de la expresión.

Formas de demostrar proposiciones:

$$\begin{aligned} \text{Directo} : p \Rightarrow q &\equiv \text{Contrarrecíproco} : q' \Rightarrow p' \\ \text{Recíproco} : q \Rightarrow p &\equiv \text{Contrario} : p' \Rightarrow q' \end{aligned}$$

1.2. Álgebras de Boole

Un álgebra de Boole se estructura como la séxtupla $(B, +, \cdot, ', 0, 1)$ siendo $+$ la suma, \cdot el producto, $'$ el complemento, 0_B el elemento neutro de la suma booleana y 1_B el elemento neutro del producto booleano.

$$\begin{aligned} x + \bar{x} &= 1_B \\ x \cdot \bar{x} &= 0_B \\ x + x &= x \cdot x = x \\ x \cdot y &= 1_B \leftrightarrow x = 1_B \wedge y = 1_B \end{aligned}$$

Dado un álgebra de Boole, se define un subálgebra de Boole $(A, +, \cdot, ', 0_B, 1_B)$ $\forall x, y \in A \wedge A \subset B$ si y solo si: $x + y \in A, x \cdot y \in A, \bar{x} \in A \forall x \in A$

Sea un álgebra de Boole $(A, +, \cdot, ', 0, 1)$; el producto cartesiano se define como: $A \times A = \{(x, y) / x \in A \wedge y \in A\}$

Si $A = \{a, b, c\}$ entonces todas las posibles relaciones binarias $(x \mathcal{R} y)$ con $x, y \in A$ pueden ser expresadas con el conjunto \mathcal{R}_n tal que $\mathcal{R}_1 = \emptyset$, $\mathcal{R}_2 = \{(a, a)\}$, $\mathcal{R}_3 = \{(a, a), (a, b)\}$, \dots , $\mathcal{R}_n = \{A \times A\}$. Notamos que $a \mathcal{R} b$ es equivalente a decir $(a, b) \in \mathcal{R}$.

La matriz de Adyacencia, Mr , de una relación, \mathcal{R} , se define como la matriz compuesta por elementos del conjunto binario $\{0, 1\}$. $Mr(i, j) = 1$ si $\exists i \mathcal{R} j$ y $Mr(i, j) = 0$ si $\nexists i \mathcal{R} j$. Además, Mr^T se define de la siguiente forma: $Mr^T(i, j) = 1$ si $Mr(i, j) = 0$ y viceversa $\forall i \neq j$. Particularmente $Mr^T(i, i) = 0$ si $Mr(i, i) = 0$ y $Mr^T(i, i) = 1$ si $Mr(i, i) = 1$.

$$\begin{aligned} \mathcal{R} \circ \mathcal{S} &= M_S \cdot Mr = \mathcal{S}^{-1} \circ \mathcal{R}^{-1} \\ (\mathcal{R} \circ \mathcal{S})^{-1} &= Mr^{-1} \cdot M_S^{-1} \\ (\mathcal{R} + \mathcal{S})^{-1} &= \mathcal{R}^{-1} + \mathcal{S}^{-1} \end{aligned}$$

Propiedades de las relaciones ($A \neq \emptyset$) :

Propiedad reflexiva : $\forall x \in A$ se cumple que $x \mathcal{R} x$. Sea Mr la matriz de adyacencia de la relación; la cual será reflexiva si $I \leq Mr$. Finalmente, si

tenemos el grafo que representa las relaciones, el mismo será reflexivo si \exists loop $\forall x \in \mathcal{R}$

Propiedad simétrica : $\forall x, y \in A$ si $\exists x\mathcal{R}y \Rightarrow \exists y\mathcal{R}x$. Sea Mr la matriz de adyacencia de la relación; la cual será simétrica si $Mr = Mr^T$. Finalmente, si tenemos el grafo que representa la relación, la misma será simétrica si se cumple que $(\forall x, y \in \text{grafo de la relación})$ para toda arista que va de x hacia y , \exists arista que va de y hacia x .

Propiedad antisimétrica : $\forall x, y \in A$ si $\exists x\mathcal{R}y \wedge y\mathcal{R}x \Rightarrow x = y$. Sea Mr la matriz de adyacencia de la relación; la cual será antisimétrica si $Mr \cdot Mr^T \leq I$ (no es producto de fila por columna, es producto de elemento a elemento). Finalmente, si tenemos el grafo que representa la relación, la misma será antisimétrica si se cumple que $(\forall x, y \in \text{grafo de la relación})$ para toda arista que va de x hacia y , \nexists arista que va de y hacia x .

Propiedad transitiva : $\forall x, y, z \in A$ si $\exists x\mathcal{R}y \wedge y\mathcal{R}z \Rightarrow x\mathcal{R}z$. Sea Mr la matriz de adyacencia de la relación; la cual será transitiva si $Mr \cdot Mr = Mr^2 \leq Mr$ (es producto de fila por columna). Finalmente, en el grafo de la relación, se dirá que el mismo es transitivo si \forall camino de x hacia y hacia z , podemos encontrar un camino directo de x a z .

Si la relación cumple con las reflexividad, la simetría y la transitividad; se dice que esta es una *relación de equivalencia*. En cambio, si la relación cumple con la reflexividad, la antisimetría y la transitividad; se dice que esta es una *relación de orden*.

Se conoce a la relación $x \preceq y$ como la que existe cuando $x \cdot y = x$.

Un átomo (a) , tal que $a \in B \wedge a \neq 0_B$, cumple una de estas dos condiciones $(\forall x \in B)$:

$$\begin{aligned} a \cdot x &= a \\ a \cdot x &= 0_B \end{aligned}$$

1.3. Isomorfismos entre Álgebras de Boole

Sea $AB_1 = (B_1, +_1, \cdot_1, \neg_1, 0_1, 1_1)$ y $AB_2 = (B_2, +_2, \cdot_2, \neg_2, 0_2, 1_2)$, f es un isomorfismo de $B_1 \rightarrow B_2$ biyectivo cuando:

$$\begin{aligned}
&\forall x, y \in B_1, f(x +_1 y) = f(x) +_2 f(y) \\
&\forall x, y \in B_1, f(x \cdot_1 y) = f(x) \cdot_2 f(y) \\
&\forall x \in B_1, f(\neg x) = \neg f(x) \\
&f(0_1) = 0_2 \\
&f(1_1) = 1_2 \\
&\text{Si } x \preceq y \Rightarrow f(x) \preceq f(y) \\
&\text{Si } a \text{ es un átomo, } f(a) \text{ también lo es, tal que } a \in B_1 \wedge f(a) \in B_2.
\end{aligned}$$

Si tenemos n cantidad o número de átomos, diremos que $\exists n!$ posibles isomorfismos entre otras álgebras de Boole.

La clausura transitiva es el mínimo conjunto de elementos que hace que \mathcal{R} sea transitiva.

1.4. Relaciones de Orden

Para toda relación de orden, su grafo representante de la relación puede ser representado por un *Diagrama de Hasse*. El mismo se representa igual que el grafo de la relación, pero en esta borramos las reflexividades y las transitividades "de menor largo".

Se dice que el conjunto está totalmente ordenado si en su Diagrama de Hasse, el grafo resultante es uno conexo (es decir existe un camino entre todos los vértices del mismo) y si no posee ciclos (que es equivalente a decir que para todo par de vértices pertenecientes al grafo, siempre existe solo un camino que los conecta). De lo contrario este estará parcialmente ordenado.

Sea H el diagrama de Hasse de la relación de orden y H_1 , un subgrafo tal que $H_1 \in H$. Se podrán notar las siguientes características de H_1 :

Propiedades varias :

Cotas superiores: Es el conjunto compuesto por los vértices que están "sobre" del conjunto de vértices de H_1 y el máximo de H_1 .

Cotas inferiores: Es el conjunto compuesto por los vértices que están "debajo" del conjunto de vértices de H_1 y el mínimo de H_1 .

Supremo : la menor cota superior.

Ínfimo : la mayor cota inferior.

Maximal : El conjunto de valores que "se encuentran arriba de todo" componen el maximal. Si este conjunto tiene solo un elemento, se dice que el mismo

es tanto maximal como *máximo*.

Minimal : El conjunto de valores que "se encuentran abajo de todo" componen el minimal. Si este conjunto tiene solo un elemento, se dice que el mismo

es tanto minimal como *mínimo*.

Las cotas inferiores, cotas superiores, supremos y mínimos existen en H , en cambio el máximo, mínimo, maximal y minimal existen en H_1 .

1.5. Relaciones de Equivalencia

Las clases de equivalencia, que se pueden notar como: $[a], \bar{a}, cl(a)$ y se define el conjunto de la siguiente forma: $[a] = \{x \in A / x \mathcal{R} a\}$

Una partición de A , A_i cumple que:

$$\begin{aligned} A_i &\subset A \\ A_i &\neq \emptyset \\ A_i \cap A_j &= \emptyset \\ A_1 \cup A_2 \cup \dots \cup A_i &= A \end{aligned}$$

Un *conjunto* cociente, $\frac{A}{\mathcal{R}}$, define una partición de A , y está compuesta por la mínima cantidad de clases de equivalencia que conforman al mismo. Se eliminan conjuntos igualmente definidos.

Si decimos que $a|b$ entonces a divide a b , y b es múltiplo de a . Además notamos la siguiente definición:

$$a = b(mod\ n) \Leftrightarrow n|(a - b) \Leftrightarrow a = kn + b.$$

1.6. Autómatas

Se define un automata finito determinista como $M = (\Sigma, \mathcal{Q}, q_0, \Gamma, \mathcal{F})$. Donde Σ , sucesión finita de caracteres, es el alfabeto; Σ^* el conjunto de todas las posibles cadenas definidas sobre Σ ; Σ^n el conjunto de cadenas de longitud n sobre Σ ; \mathcal{Q} el conjunto de estados (es decir "los vértices del autómata"); q_0 el estado inicial; Γ la función de transición $\mathcal{Q} \times \Sigma \rightarrow \mathcal{Q}$; \mathcal{F} el conjunto de aceptación, $L(M)$ el lenguaje del autómata, definido de la forma: $\{x \in \Sigma^* / \Gamma^*(q_0, x) \in \mathcal{F}\}$, con $\Gamma^* : \mathcal{Q} \times \Sigma^* \rightarrow \mathcal{Q}$; finalmente la expresión: $|x|_a = n$ es equivalente a decir, que la sucesión de palabras, debe contener n letras a .

2. Trabajo Práctico II

2.1. Inducción Matemática

Sea $P(n)$ una propiedad que se define en el conjunto de \mathbb{N} . Esta propiedad se demostrará por inducción de la siguiente manera:

1. Se debe demostrar que $P(n_0)$ es verdadera, siendo $n_0 \in \mathbb{N}$ el primer elemento del conjunto para la cual la propiedad se cumple.
2. Asumimos que $P(n)$ es verdadera $\forall n \geq n_0$ y con $n \in \mathbb{N}$.
3. Si se demuestra que $P(n) \Rightarrow P(n+1)$, la propiedad será verdadera $\forall n \in \mathbb{N}$.

Esto solo funciona para conjuntos bien ordenados, es decir, que tienen un elemento mínimo.

2.2. Relaciones de Recurrencia

Sea $X_n : \mathbb{N}_0 \rightarrow \mathbb{R}$, diremos que la relación de recurrencia de primer orden tiene la siguiente forma: $X_{n+1} = A_n \cdot X_n + B_n$.

Sea X_0 la condición inicial; se puede expresar a X_n de la siguiente forma:

$$\begin{aligned} X_1 &= A_0 \cdot X_0 + B_0 \\ X_2 &= A_1 \cdot X_1 + B_1 = A_1 \cdot (A_0 \cdot X_0 + B_0) + B_1 \\ &\vdots \\ X_n &= X_0 \cdot \prod_{i=1}^{n-1} A_i + \sum_{k=0}^{n-1} \left(\prod_{i=k+1}^{n-1} A_i \right) \cdot B_k \end{aligned}$$

Sea $X_n : \mathbb{N}_0 \rightarrow \mathbb{R}$, diremos que la relación de recurrencia lineal homogénea de segundo orden tiene la siguiente forma: $X_{n+2} + C_{n+1} \cdot X_{n+1} + C_n \cdot X_n = 0$. Las soluciones del mismo tienen la forma de $X_n = c \cdot R^n$ y con $c \neq 0, R \neq 0$. Al ser de segundo orden, necesito dos condiciones iniciales para poder conseguir el resultado exacto.

Al reemplazar $X_n = c \cdot R^n$ en nuestra ecuación de segundo orden, obtendremos el polinomio característico igualado a cero, el cual nos dará las soluciones para R_1 y R_2 . Notamos que $R_1, R_2, \dots, R_n \in \mathbb{C}$.

Recordamos brevemente que:

$$z^n = |z|^n \cdot (\cos(n \cdot \alpha) + i \cdot \sen(n \cdot \alpha))$$

$$\bar{z} = |z| \cdot (\cos(-\alpha) + i \cdot \sen(-\alpha))$$

$$\cos(\alpha) = \cos(-\alpha)$$

$$\sen(\alpha) = -\sen(-\alpha)$$

Si de una ecuación de recurrencia conseguimos la solución:

$X_n = c_1 \cdot R^n + c_2 \cdot R^n + c_3 \cdot R^n$ y sus términos no son linealmente independientes entre sí, multiplicaremos por n^k a estos para que si lo sean, y la solución ahora sería: $X_n = c \cdot R^n + (c \cdot R^n) \cdot n + (c \cdot R^n) \cdot n^2$.

Sea $X_n : \mathbb{N}_0 \rightarrow \mathbb{R}$, diremos que la relación de recurrencia no homogénea tiene la forma: $X_n + C_{n-1} \cdot X_{n-1} + C_{n-2} \cdot X_{n-2} + \dots + C_{n-k} \cdot X_{n-k} = f(n)$. Las soluciones del mismo tienen la forma de $X_n = X_n^h + X_n^p$ siendo X_n^h la solución del homogéneo y X_n^p la solución del particular.

Los pasos para resolver esta ecuación no homogénea son:

1. Propongo la solución del particular, la cual obtengo de la tabla de soluciones, correspondiente a la $f(n)$ correcta.
2. Consigo $X_n^h = X_n = c \cdot R^n$ igualando la parte homogénea a cero, y por separado sustituyo X_n^p en la parte homogénea y lo igualo a $f(n)$ para obtener X_n^p .
3. Ahora con las condiciones iniciales despejamos $X_n = X_n^h + X_n^p$.

$f(n)$	X_n^p
a (cte)	c (cte)
n	$a_1 \cdot n + a_0$
n^t	$a_t \cdot n^t + a_{t-1} \cdot n^{t-1} + \dots + a_0$
a^n	$c \cdot a^n$
$\sen(\alpha) \vee \cos(\alpha)$	$A \cdot \sen(n \cdot \alpha) + B \cdot \cos(n \cdot \alpha)$
$a^n \cdot n^t$	$a^n \cdot (a_t \cdot n^t + a_{t-1} \cdot n^{t-1} + \dots + a_0)$
$a^n \cdot (\sen(\alpha)) \vee a^n \cdot (\cos(\alpha))$	$a^n \cdot (A \cdot \sen(n \cdot \alpha) + B \cdot \cos(n \cdot \alpha))$
$n + a$	$(n \cdot b + c) \cdot n$

La sucesión de Fibonacci es: $F_n = F_{n-1} + F_{n-2}$.

Y su solución es: $F_n = \frac{\varphi^n - (1 - \varphi)^n}{\sqrt{5}}$, $\varphi = \frac{1 + \sqrt{5}}{2}$.

Torres de Hanói: $H_n = 2H_{n-1} + 1$.

3. Trabajo Práctico III

3.1. Introducción a la Teoría de Grafos

En teoría de grafos, diremos que $G = (V(G), E(G))$ siendo $V(G)$ el conjunto de vértices del grafo, $E(G)$ el conjunto de aristas del grafo, $|V(G)| = n$ es el *orden* (cantidad de vértices) y $|E(G)| = m$ es el *tamaño* (cantidad de aristas).

Un *camino* se define como una sucesión de vértices y aristas. Un *trail* es un camino que no repite aristas. Un *path* es un camino que no repite vértices. Un *ciclo* (C_n) es un camino cerrado, sin vértices repetidos (excepto por el primer y último vértice). Un *circuito* es un camino cerrado sin aristas repetidas.

$$K(C_n) = 2, \text{ si } n \text{ es par } K(C_n) = 3, \text{ si } n \geq 3 \text{ es impar}$$

Un *grafo camino* (P_n), es cuyo path máximo es igual al mismo grafo, es decir, el grafo es el path máximo.

$$K(P_n) = 2, \forall n \geq 2$$

El *degree* o *grado* de un vértice, es la cantidad de aristas que inciden en el mismo. Por definición, en un loop la arista que se conecta a al mismo vértice añade un grado extra.

$$\sum_{k=0}^n d(v_k) = 2m$$

Degree Sequence: "conjunto" formado por todos los grados de todos los vértices del grafo, ordenados de menor a mayor. Se escribe usando paréntesis.

$$\text{Grado mínimo: } \delta(G) = \min_v(d(v)).$$

Grado máximo: $\Delta(G) = \max_v(d(v))$.

Excentricidad: la máxima distancia de i a j ; $e(G) = \max_{v_i}(d(v_i, v_j))$.

Diámetro $\phi(G)$: la máxima excentricidad del grafo.

Radio $R(G)$: la mínima excentricidad del grafo.

$$R(G) \leq \phi(G) \leq 2R(G)$$

Centro: conjunto de vértices de excentricidad mínima.

Periferia: conjunto de vértices de excentricidad máxima.

Perímetro: longitud mínima de un ciclo.

Propiedades para las distancias entre vértices ($\forall u, v, w \in V(G)$):

$$d(u, v) \geq 0$$

$$d(u, v) = 0 \Rightarrow u = v$$

$$d(u, w) = d(u, v) + d(v, w)$$

$$\nexists \text{ camino entre } u \text{ y } v \Leftrightarrow d(u, v) = \infty$$

Grafo regular: $\forall v \in V(G), d(v) = k$

Grafo completo K_n : siempre existe un camino que conecta a todo par de vértices pertenecientes al grafo.

$$\forall v \in K_n, d(v) = n - 1$$

$$m = \frac{n!}{2!(n-2)!} = \binom{n}{2}$$

K_n es planar sii $n \leq 5$

$$\sigma(K_n) = \{n-1, 1-n\}$$

Sea $G = (V(G), E(G))$ y G' su *complemento*, tal que $G' = (V(G'), E(G'))$:
 $V(G') = V(G) \wedge E(G') = E(K_n) - E(G)$. G y G' *jamás* pueden ser conexas

a la vez.

Isomorfismo: propiedad o relación entre grafos que comparten elementos particulares como $\Delta(G)$, $\delta(G)$, $R(G)$, $\phi(G)$. Si G y H son isomorfos entonces: $G \cong H$. Formalmente hablando, el isomorfismo entre grafos existe al demostrarse que $\forall v \in V(G) \exists \Phi : V(G) \rightarrow V(H)$, con Φ biyectiva. Se debe cumplir que $\forall e_1 \in E(G)$ con $e_1 = \{u, v\}$, $\exists e_2 \in E(H)$ tal que $e_2 = \{\Phi(u), \Phi(v)\}$, con $u, v \in V(G)$ y $\Phi(u), \Phi(v) \in V(H)$. Los isomorfismos preservan las k -regularidades y sucesiones gráficas.

G_2 es un *subgrafo* de G_1 si $V(G_2) \subset V(G_1)$ y $E(G_2) \subset E(G_1)$.

Tree: grafo conexo sin ciclos, tal que $m = n - 1$.

Sea A la *matriz de adyacencia* de G . $A_{ij} = 1$ si $\exists e \in E(G)$ tal que $e = \{v_i, v_j\}$. $A_{ij} = 0$ en caso contrario. Además sea A^k ; entonces A_{ij}^k representa la cantidad de caminos de longitud k que existen entre v_i y v_j .

Grafo autocomplementario: isomorfo a su complemento. $n = 0 \pmod{4}$ o $n = 1 \pmod{4}$.

Grafo simple: no posee loops, y solo existe como máximo una arista que conecte a dos vértices.

$$m \leq \binom{n}{2}$$

$$d(v_k) = A_{kk}^2$$

$$2m = \sum_{i=1}^n \lambda_i^2$$

$$6\tau = \sum_{i=1}^n \lambda_i^2$$

Grafo euleriano: es el grafo conexo tal que posee un path cerrado que contiene a todas sus aristas sin repetirlas. Es *semieuleriano* si el path no es cerrado. Para que un grafo bipartito completo sea euleriano, n debe ser par. $\forall v \in V(G), d(v) = 0 \pmod{2}$.

Grafo hamiltoniano: es el grafo conexo tal que posee un path cerrado que contiene a todos sus vertices sin repetirlos. Ningún grafo bipartito es hamil-

toniano si n es impar.

Teorema de Dirac: Sea G un grafo conexo, simple y sin lazos; si $\delta(G) \geq 2$, entonces G es hamiltoniano.

Teorema de Ore: Sea G un grafo conexo, simple y sin lazos; $\forall v, u \in V(G)$ si $d(u) + d(v) \geq n$ entonces G es hamiltoniano.

Teorema de Dirac \subset Teorema de Ore.

Para todo grafo planar se cumple que: $n - m + f = 2$.

Un *grafo bipartito* cumple que para X e Y ($X \cup Y = V(G) \wedge X \cap Y = \emptyset$), $\nexists e = (x, y)$ con $x \in X, y \in Y \wedge e \in E(G)$.

$K_{n,n}$ es planar sii $n < 3$.

$$\sigma(K_{p,q}) = \{\sqrt{pq}, -\sqrt{pq}, 0(p+q-2)\}$$

Grafo planar: representable graficamente de tal forma que ninguna arista intersecte a otra. Todo grafo planar posee un árbol generador y $m \leq 3(n-2)$.

Bosque F : conjunto de árboles conexos. $K(F) = 2$

Caras: "espacios" generados producto del encierro de aristas. Su grado, o *face degree* $d(f)$, equivale a la cantidad de caras que componen los bordes de la misma. Si $d(f) = 3$ la cara es un triángulo, y se lo denomina como: τ . Además si $d(f) \geq 3 \Rightarrow 2m \leq 3f, \forall f \in F(G)$.

Un *grafo dual* G^* de un G planar se obtiene conectando los vértices de G^* sii las caras correspondientes "se tocan" en G . $\forall G^*$ se cumple que: $\{V(G^*) = F(G), F(G^*) = V(G), E(G^*) = E(G)\}$. Un *grafo autodual* es isomorfo a su dual: $G^* \cong G$.

Sea G sin loops, este estará k -colorado si para cada vértice de G se lo puede pintar de un color tal que los vértices adyacentes tengan siempre un color diferente; la k representa la cantidad de colores. De aquí sale la definición del número cromático $K(G)$ que representa el mínimo número de colores, k , que cumple la propiedad. Para la coloración de aristas usamos $K'(G)$.

Grafo línea $L(G)$: Sea G , diremos que: $V(L(G)) = E(G)$ y las aristas de $L(G)$ se conectan a los vértices que, en su forma de arista en G .

$$K_n \subset G \text{ sii } K(G) \geq n$$

$$K(K_n) = n$$

$$\forall G \text{ bipartito, } K(G) = 2$$

$$K(G) \leq \Delta(G) + 1 \leq K'(G) + 1$$

Sea A la matriz de adyacencia de G . Los autovalores de A son los autovalores de G . Los autovalores se consiguen con la operación $\det(A - \lambda I) = 0$.

$$\text{Sea } u, v \in V(G), w \in E(G), w \in V(L(G)) \Rightarrow d(w) = d(u) + d(v) - 2.$$

$$G \cong L(G) \text{ sii } G \text{ es 2-regular.}$$

$$E(L(G)) = \frac{1}{2} \sum_{k=1}^n d(v_k)^2 - E(G) = \sum_{k=1}^n \binom{d(v_k)}{2}$$

Espectro de un grafo $\sigma(G)$: es el conjunto de los autovalores de G .

$$\sum_{k=1}^n \lambda_k = \text{tr}(A) = 0$$

$$\text{Si } \lambda \in \sigma(B) \Rightarrow \lambda^q \in \sigma(B^q).$$

$$\text{Si } \lambda \in \sigma(B) \Rightarrow \lambda - a \text{ es autovalor de } A = B - aI.$$

$$\text{Si } G \cong H \Rightarrow \sigma(G) = \sigma(H)$$

$$\sigma(P_n) = \{2\cos(\frac{k\pi}{n+1})\}, \forall k \in \mathbb{N}$$

$S \in V(G)$ es un *corte de vértices* si $H_1 = (V(G) - S, E(H))$ es conexa y $T \in E(G)$ es un *corte de aristas* si $H_2 = (V(H), E(G) - T)$ es conexa.

La *vértice conectividad* $\kappa(G)$ es el mínimo cardinal de S que haga a $H_1 = (V(G) - S, E(H))$ no conexa. En cambio la *arista conectividad* $\lambda(G)$ es el mínimo cardinal de T que hace que $H_2 = (V(H), E(G) - T)$ sea no conexa. Además si G es conexa; $\delta(G) \geq \lambda(G) \geq \kappa(G)$; y si $a \in V(G)$, a es

un *bridge* si $\kappa(G) = 1$ con $S = \{a\}$.

Un *path maximal* es el que no puede ser incluido en otro path.

Sean $u, v \in V(G)$ tal que existe un camino entre ellos, diremos que estos caminos son *arista-disjuntos* si no comparten aristas y diremos que estos caminos son *vértice-disjuntos* si no comparten vértices (excepto por el inicial y el final).

La *matriz de incidencia* de G tiene columnas que representan las aristas y las filas representan el vértice al que se conectan. En la de adyacencia obviamente tanto las filas como columnas representan vértices.

Una *hoja*, o *leaf*, son vértices de grado 1. Además, $\forall T$ tal que $|V(T)| \geq 2$ tiene al menos dos hojas.

3.2. Grafos Ponderados

Un *grafo ponderado*, se diferencia de un grafo cualquiera ya que cada una de sus aristas tiene un peso asignado. Hasta ahora, el peso de toda arista fue 1, pero ahora el valor puede variar. Un ejemplo muy común del uso de estos grafos, es representar ciudades con vértices y el peso de cada arista representa la distancia del camino que los conecta, en caso de que este exista. Finalmente, sea A la matriz de adyacencia del grafo ponderado G ; ahora $A_{ij} = W_k$, siendo W_k el peso de la arista que conecta a v_i con v_j .

$$W(T) = \sum_{k=1}^{n-1} W_k, \text{ siendo } W_k \text{ el peso de la } k\text{-ésima arista de } T.$$

Árbol generador del peso mínimo: sea G , queremos encontrar un árbol T , tal que pueda generar a G . Además este debe tener el peso mínimo, es decir, $\forall T_i, W(T) \leq W(T_i)$.

Algoritmo de Kruskal: permite obtener un árbol generador usando la representación gráfica del mismo. Se deben seguir los siguientes pasos:

1. Comprobar que para G , conexa, $\exists T \subset G$.
2. Si es posible iniciar el algoritmo en el vértice con la arista de menor peso.
3. Viajar por la arista de menor peso posible.

4. Repetir el paso anterior, **sin crear ciclos**, teniendo en cuenta a todos los vértices a los que ya cubrimos, hasta haber llegado a todos los vértices de G .

Algoritmo de Prim: permite obtener un árbol generador usando la matriz de adyacencia del mismo. Se deben seguir los siguientes pasos:

1. Comprobar que para G , conexa, $\exists T \subset G$; sea A la matriz de adyacencia de G .
2. Elegir un vértice cualquiera.
3. En la columna de A que corresponde al vértice elegido, buscar el menor peso bajo las columnas seleccionadas.
4. Elegir esta y volver a repetir el paso anterior, solo que ahora eliminamos la fila de los vértices ya elegidos y añadimos la columna del vértice elegido a las que usamos para encontrar los pesos mínimos. Repetimos hasta que todas las filas de A hayan sido eliminadas.

Algoritmo de Dijkstra: calcula desde un vértice a hasta un vértice b el camino más corto de menor peso. Se inicia en a e iteradamente se salta al vértice adyacente por la arista de menor peso; así hasta llegar a b .

3.3. Grafos Orientados

Este tipo de grafos poseen aristas con flechas, que indican la dirección del flujo. Son el tipo de grafos que usamos para estudiar autómatas y los que aparecían en la unidad de relaciones.

Supongamos que tenemos el grafo G_1 no orientado o dirigido. Este solo posee dos vértices y una arista que los conecta. Sabemos que $d(v_1, v_2) = d(v_2, v_1) = 1$. En cambio sea el grafo G_2 dirigido. Este solo posee dos vértices y una arista que los conecta *pero esta va de v_1 hacia v_2* . En este caso $d(v_1, v_2) = 1 \neq d(v_2, v_1) = \infty$.

Grafo fuertemente conexo: $\forall v, u \in V(G)$, \exists camino de v a u y viceversa.

$$d^+(v) = \text{in-degree ("aristas que ingresan")}$$

$$d^-(v) = \text{out-degree ("aristas que salen")}$$

$$d(v) = d^+(v) + d^-(v)$$

$$\sum_{i=1}^n d^-(v_i) = \sum_{j=1}^n d^+(v_j) = m$$

Breadth First Search (BFS): algoritmo que transforma un grafo normal, en orientado y fuertemente conexo.

1. Tomo un vértice cualquiera.
2. Exploro todos sus adyacentes, recordando el orden de descubrimiento.
3. Respetando el orden antes impuesto, investigamos uno por uno los vértices adyacentes de los anteriormente encontrados; y así hasta llegar a todos los vértices del grafo.

Depth First Search (DFS): algoritmo que transforma un grafo normal, en orientado y fuertemente conexo.

1. Tomo un vértice cualquiera.
2. Exploro los adyacentes, e intento ir por el camino más extenso posible, cubriendo tantos vértices como sea posible sin crear ciclos.
3. Vuelvo para atrás si es necesario para extenderme a otras direcciones y llegar a los vértices restantes. Así hasta llegar a todos.
4. Completar las aristas restantes de tal forma que $\forall v, u \in V(G)$,
 \exists camino de v a u y viceversa.

Torneo: grafo orientado o digrafo, que al sacarle las orientaciones es conexo. Todo torneo tiene al menos una orientación fuertemente conexa.

Algoritmo de Ford-Fulkerson: Busca transportar la máxima cantidad de algo, desde una fuente a un sumidero. Cada arista tiene una capacidad máxima, y comienza con flujo nulo. Para que el flujo sea máximo se debe intentar saturar la máxima cantidad de caminos de la mejor forma posible, cosa que depende del grafo con el que estemos trabajando.