1. Conjuntos, Subconjuntos, Conjunto Potencia y formas de representación.

- El conjunto S que contiene a todos los subconjuntos de un conjunto dado C, se conoce como el **conjunto potencia** de D.
- La cantidad de elementos de un conjunto cualquiera se conoce como la cardinalidad de ese conjunto.
- Si D tiene n elementos (card(D) = n) , entonces la cantidad de subconjuntos de D es igual a 2^n (todas las formas de combinar esos n elementos, más el conjunto vacío). Es decir, card(S) = 2^n.
- Ejemplo: Si C = {1, 2, 3, 4} – card(D) = n = 4, entonces el conjunto potencia S será:

S = {{}, {1}, {2}, {3, {4}, {1, 2}, {1, 3}, {1, 4}, {2, 3}, {2, 4}, {3, 4}, {1, 2, 3}, {1, 2, 4}, {1, 3, 4}, {2, 3, 4}, {1, 2, 3, 4}}

con card(S) = $2^n = 2^4 = 16$

- Una forma concisa, elegante y rápida de representar conjuntos que contienen números entre 1 y k, consiste en usar equivalencia entre cada conjunto y un valor numérico expresado en binario:

Conjunto	Binario	Decimal
{}	0000	0
{1}	0001	1
{2}	0010	2
{3}	0100	4
{4}	1000	8
{1, 2}	0011	3
{1, 3}	0101	5
{1, 4}	1001	9
{2, 3}	0110	6
{2, 4}	1010	10
{3, 4}	1100	12
{1, 2, 3}	0111	7
{1, 2, 4}	1011	11
{1, 3, 4}	1101	13
{2, 3, 4}	1110	14
{1, 2, 3, 4}	1111	15

- En la secuencia binaria, los bits de derecha a izquierda se usan para representar la pertenencia o no de cada número al conjunto. La secuencia 0101 está indicando que el 1 pertenece al conjunto, el 2 no, el 3 sí, y el 4 no. Por lo tanto, el conjunto asociado es el {1, 3}. Y como (0101)2 = (5)10 decimos que el conjunto {1, 3} queda representado por el número 5.
- Si se tiene un conjunto d así representado, la siguiente expresión agrega el valor x a ese conjunto, convirtiéndolo en el conjunto e = d $\cup$ {x}:

e = d | (1 << (x-1))

Ejemplo:

sea d = 9 = $(1001)_2$ ⇨ {1, 4}

y sea x = 3

entonces:
e = d | (1 << (x-1))
e = 9 | (1 << (3-1))
e = 9 | (1 << 2)
e = 1001 | 0100
e = 1101 ⇨ {1, 3, 4} = {1, 4} ∪ {3}

- En forma similar, si se quiere saber si un valor x pertenece a un conjunto d, se puede hacer con la siguiente expresión:

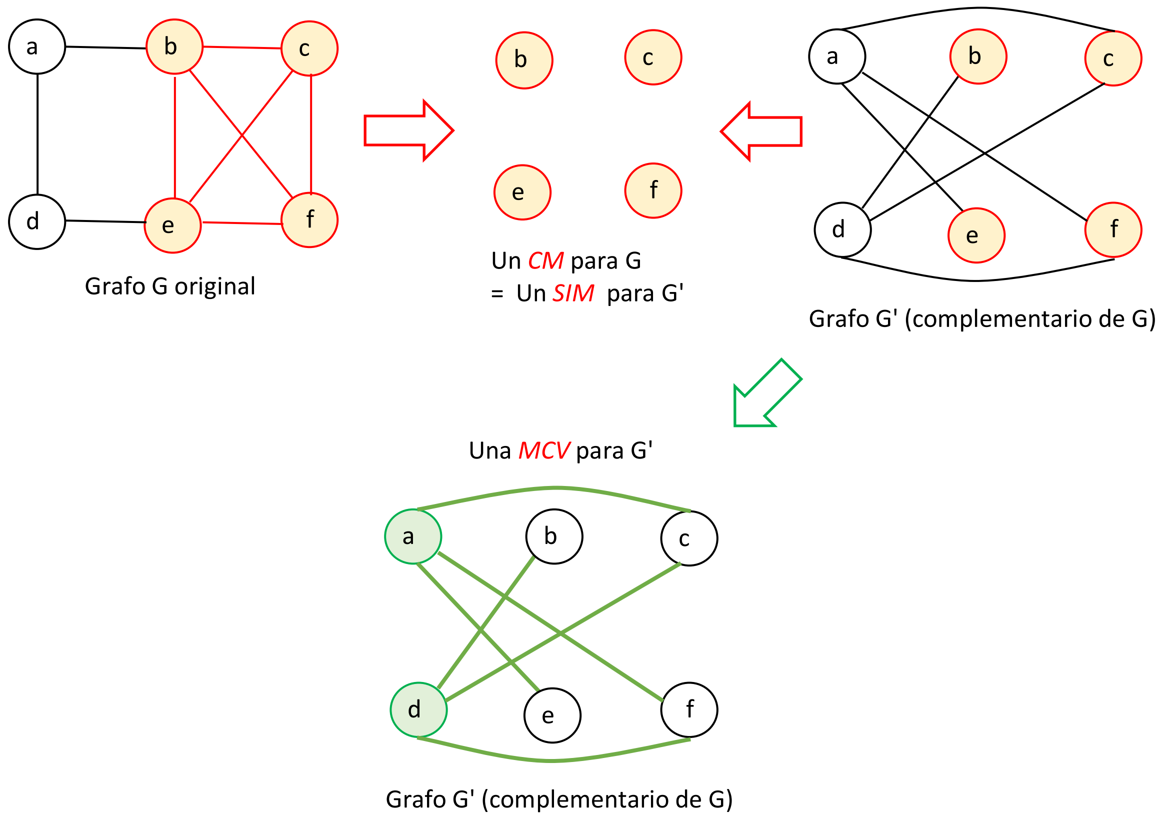
  r = d & (1 << (x-1))

  de forma que si r es 0, entonces x **no** pertenece a d.

  Ejemplo:
  sea d = 5 = $(0101)_2$ ⇨ {1, 3}
  y sea x = 2 = $(0010)_2$
  entonces:
  r = d & (1 << (x-1))
  r = 5 & (1 << (2-1))
  r = 5 & (1 << 1)
  r = 0101 & (0001 << 1)
  r = 0101 & 0010
  r = 0000 = 0
  ⇨ x = 2 ∉ d (2 no pertence a {1, 3})

2. Clique Máximo, Conjunto Independiente Máximo y Mínima Cobertura de Vértices.

- *Clique Máximo (CM)* (de un grafo G no dirigido): subconjunto de G con la mayor cantidad de vértices unidos entre sí por arcos directos (la mayor cantidad de vértices adyacentes uno a uno).
- *Subconjunto Independiente Máximo (SIM)* (de un grafo G no dirigido): subconjunto de G con la mayor cantidad de vértices NO unidos entre sí por arcos directos (la mayor cantidad de vértices NO adyacentes uno a uno).
- *Mínima Cobertura de Vértices (MCV)* (de un grafo G no dirigido): subconjunto de G con la menor cantidad de vértices tales que ***todos*** los arcos de G inciden en alguno de esos vértices.
- Estos tres problemas (hasta donde sabemos) son intratables y de hecho, son NP-Complete.
- Pueden reducirse con sencillez entre ellos, ya que se pueden probar las siguientes relaciones:
  - ✓ El CM de un grafo G es igual al SIM del grafo G' complementario de G.
  - ✓ La MCV de un grafo G es igual al subconjunto de vértices de G que NO están incluidos en un SIM para el mismo grafo G (MCV = V – SIM, siendo V el conjunto de vértices de G).
- Las siguientes gráficas aclaran las ideas:



- En definitiva:
  - ✓ Si se tiene un algoritmo para resolver cualquiera de estos tres problemas, el mismo algoritmo puede aplicarse para resolver los otros dos, haciendo simples reducciones polinómicas entre ellos.
  - ✓ Si los grafos están implementados con matrices de adyacencia, la única reducción que se requiere es convertir la matriz de un grafo en su matriz complementaria (todos los 1 de una se convierten en 0 en la otra, y se deja la diagonal principal en 0 en todos los casos).
  - ✓ Si se tiene (por ejemplo) un algoritmo para obtener la MCV de un grafo G, entonces:
    - i. Si se quiere obtener una MCV de G, se aplica ese algoritmo sin más.
    - ii. Si se quiere obtener un SIM de G, se aplica el mismo algoritmo, y luego simplemente se toman los vértices que NO están en la MCV obtenida (ni siquiera es necesario reducir la matriz).
    - iii. Si se quiere obtener un CM para G, se reduce la matriz de G para obtener G'. En G' se aplica el algoritmo que se acaba de describir para obtener el SIM, y ese mismo SIM se retorna como CM para G.
- Eso es lo que haremos en el modelo *DLC-Reductions* que presentamos para cerrar el curso… Esperamos que les sea útil.
- ¡Hasta siempre!