Teoría de Grafos III

Contenido

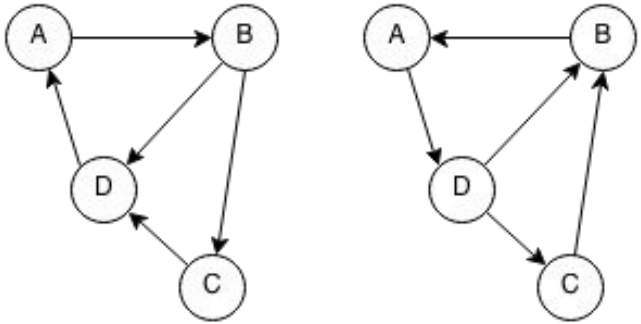
Contenido

1. Conceptos básicos	
2. Counting Reacheable Nodes Problem	
3. Grafo condensado	
4. Algoritmo de Kosaraju	
5. 2 SAT Problem	

Tipos de Grafos

Grafo Fuertemente Conexo

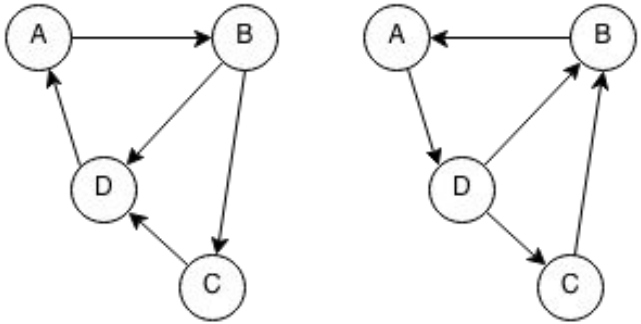
Un grafo dirigido se denomina fuertemente conexo si para cada par de vértices (u, v) existe un camino dirigido de ida (de u hacia v) y de regreso (de v hacia u).



Tipos de Grafos

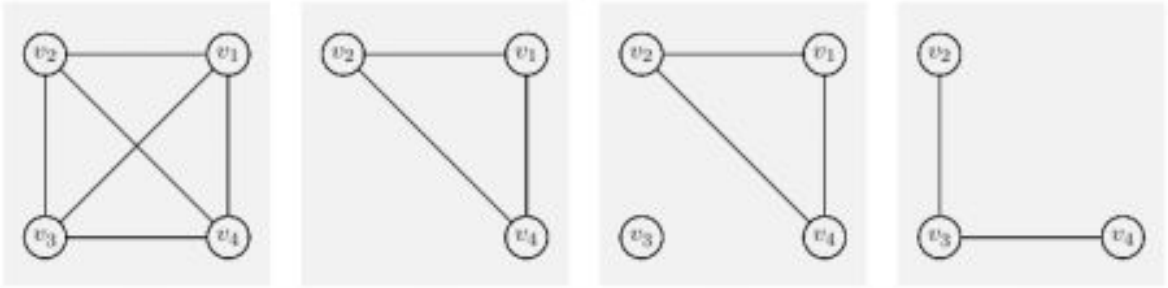
Grafo Fuertemente Conexo

Un grafo dirigido se denomina fuertemente conexo si para cada par de vértices (u, v) existe un camino dirigido de ida (de u hacia v) y de regreso (de v hacia u).



Subgrafo

Un subgrafo de un grafo $G = (V, E)$ es un grafo $H = (V_H, E_H)$ tal que $V_H \subseteq V$ y $E_H \subseteq E$

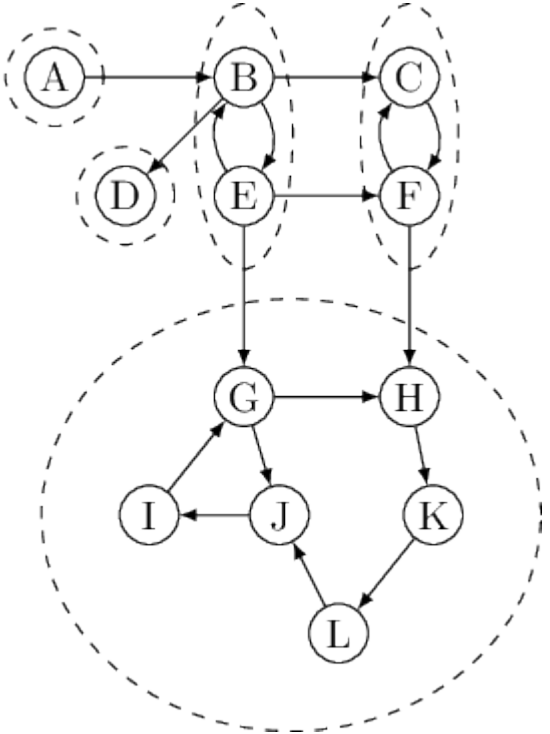


Subgrafos

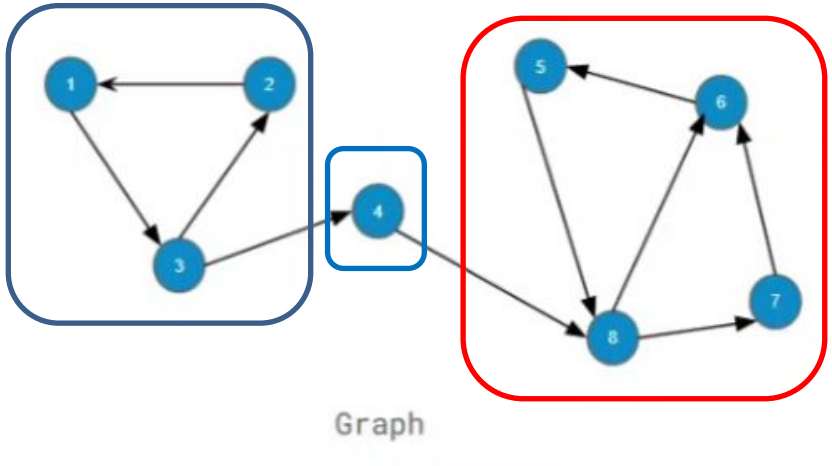
Subgrafo

Componente Fuertemente Conexa

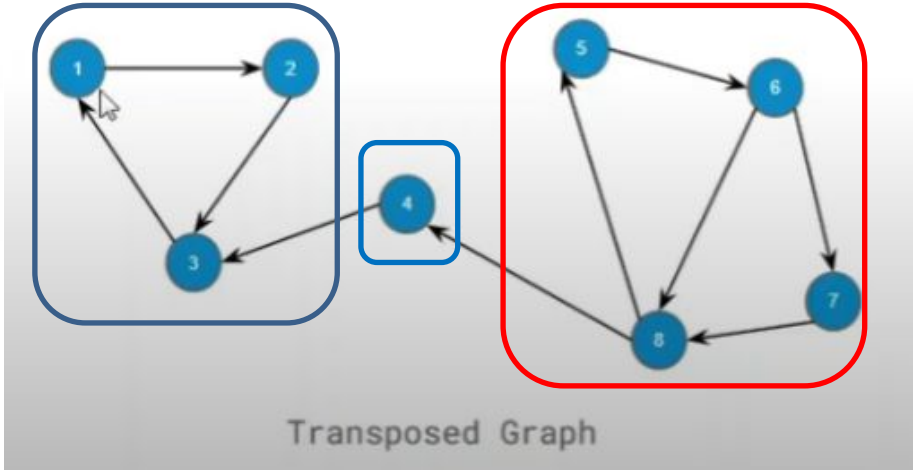
- Es un subgrafo fuertemente conexo maximal de un grafo dirigido.



Grafo Transpuesto



El grafo transpuesto se da al transponer todas las aristas de un grafo dirigido
Se observa que las componentes fuertemente conexas se mantienen al transponer el grafo.



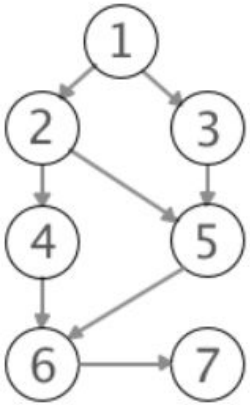
Contenido

1. Conceptos básicos	
2. Counting Reacheable Nodes	
3. Grafo condensado	
4. Algoritmo de Kosaraju	
5. 2 SAT Problem	

Problema Ejemplo: Counting Reacheable Points

- Enunciado

Dado un grafo dirigido se te pide contar la cantidad de nodos a las que se puede alcanzar siguiendo alguna secuencia de aristas. Responder este problema para todos los nodos

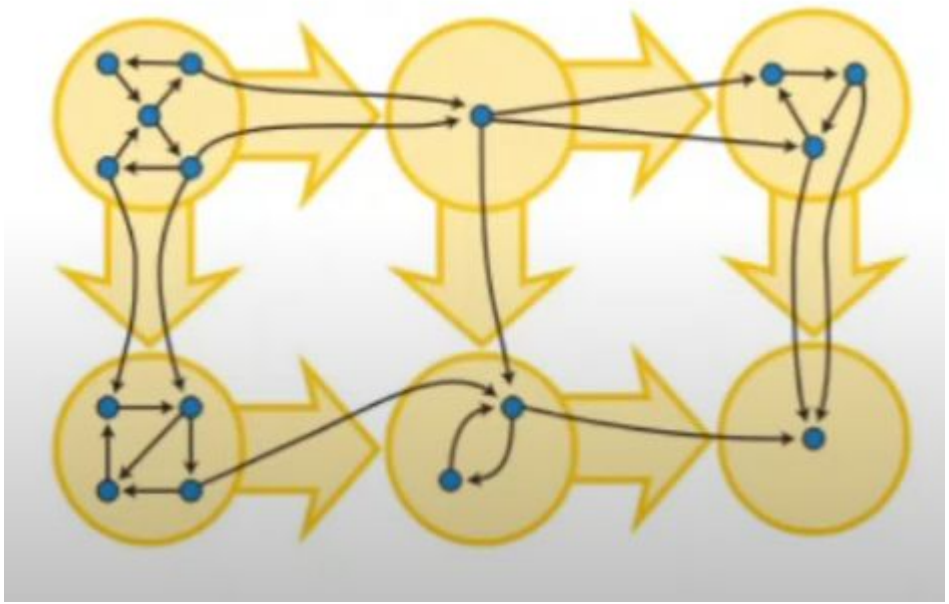


- Solución trivial

Calcular un dfs para cada nodo de forma independiente y contar todos los nodos visitados.
Complejidad O(n^2)

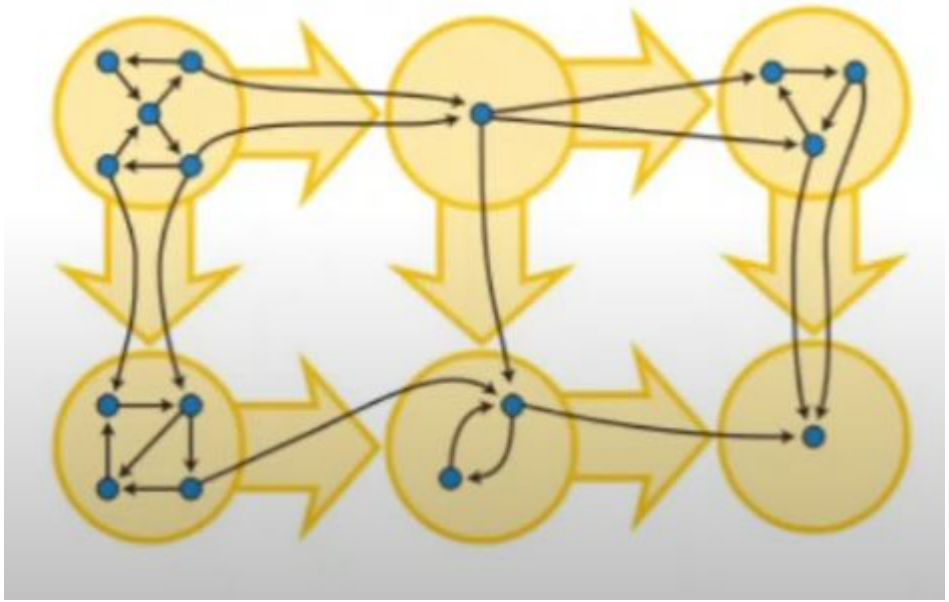
Contenido

Grafo Condensado



Es un grafo basado en las componentes fuertemente conexas del grafo original. Cada SCC es un nodo y si existe alguna arista que una algún SCCi y SCCj entonces habrá una arista entre ambos nodos.

Grafo Condensado



Propiedades:

- Grafo dirigido
- Grafo acíclico

En otras palabras, el grafo condensado en un DAG (Directed aciclic graph) por lo que se puede utilizar DP en este tipo de grafo.

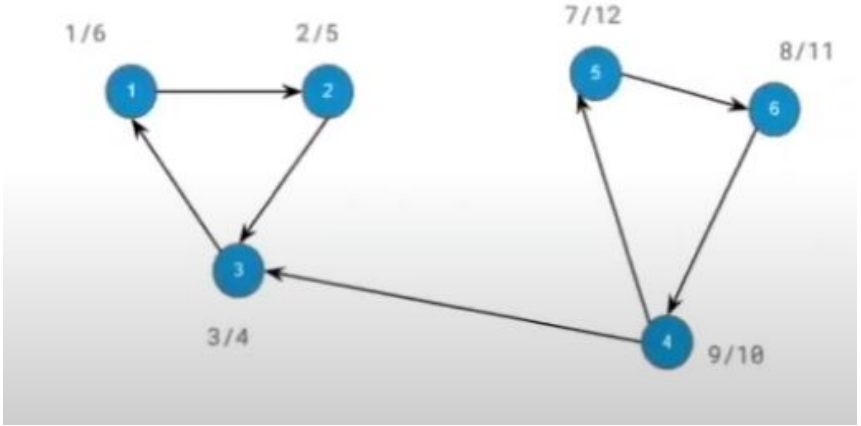
Contenido

Algoritmo de kosaraju

Algoritmo que nos identificara las componentes fuertemente conexas en un grafo dirigidos.

Principales observaciones:

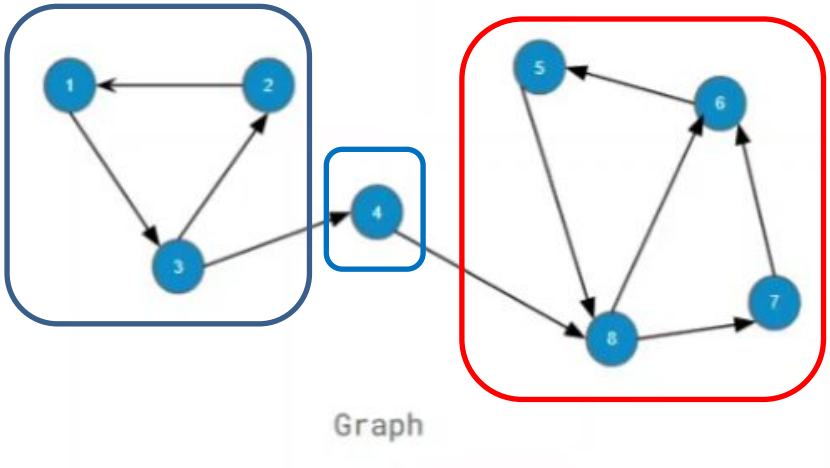
- Si existe una arista entre SCC_i y SCC_j entonces el out[SCC_i] > out[SCC_j]

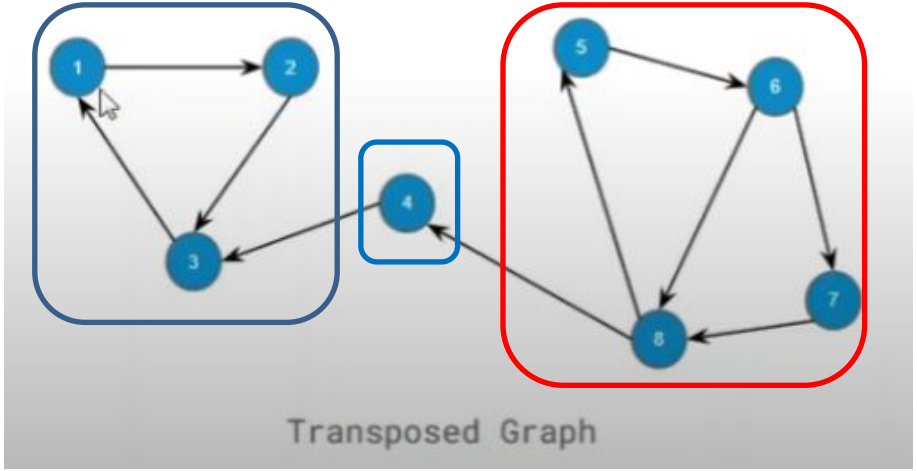


Algoritmo de kosaraju

Algoritmo que nos identificara las componentes fuertemente conexas en un grafo dirigidos.
Principales observaciones:

- En un DAG existe almenos un nodo con in-degree = 0.



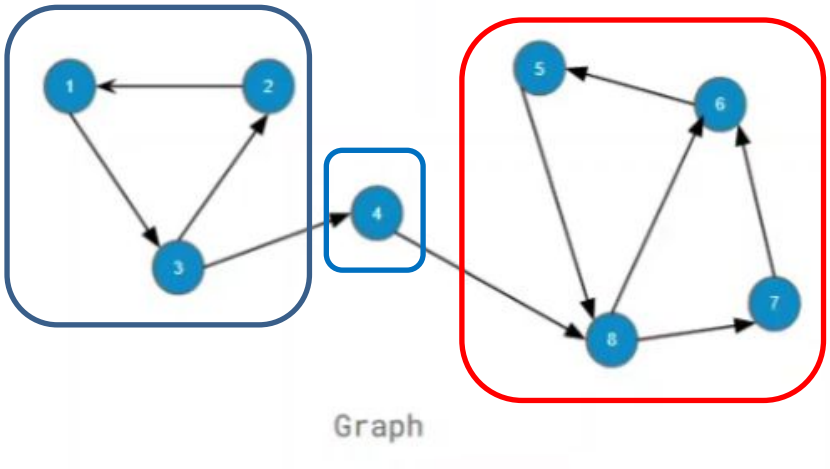


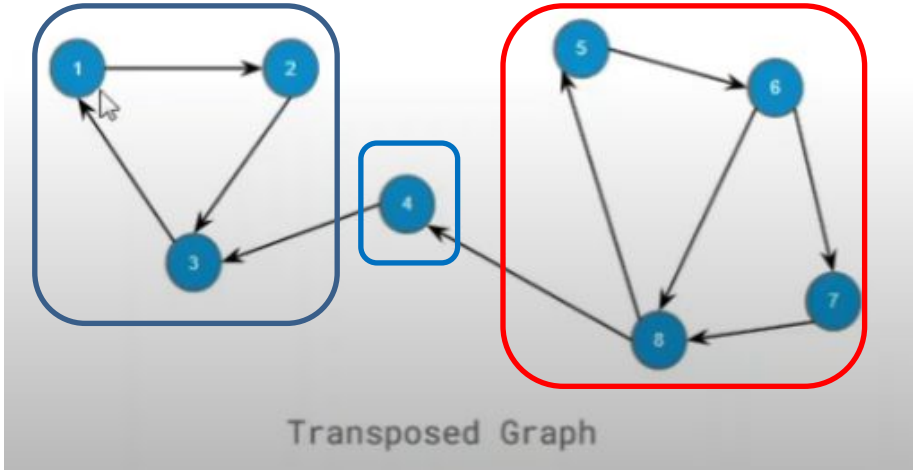
Algoritmo de kosaraju

Algoritmo que nos identificara las componentes fuertemente conexas en un grafo dirigidos.

Principales observaciones:

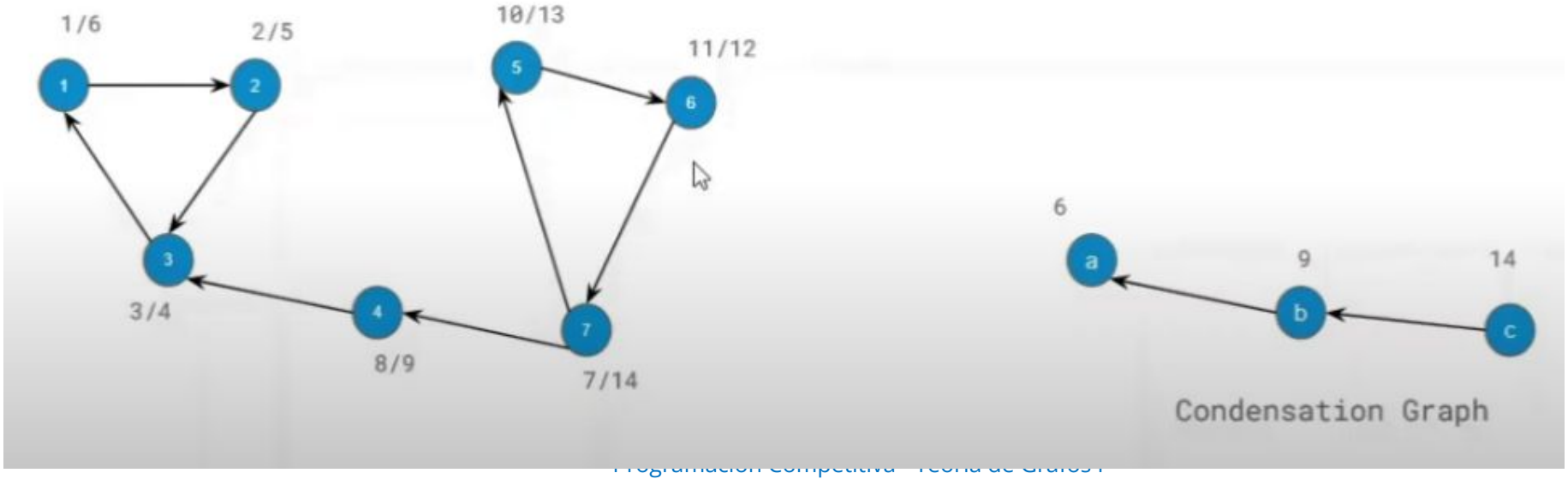
- Un nodo perteneciente al componente con in-degree 0 alcanzara solo a todos los nodos de dicha componente en el grafo transpuesto





Algoritmo de kosaraju

- Correr el DFS y ordenar los nodos por la lista de out's.
- Corre un segundo DFS sobre el grafo transpuesto en el orden de out's.
- Construir el grafo condensado recorriendo nuevamente el grafo.



Contenido

1. Conceptos básicos	
2. Counting Reacheable Nodes	
3. Grafo condensado	
4. Algoritmo de Kosaraju	
5. 2 SAT Problem	

Problema Ejemplo: 2 SAT

- Enunciado

Dado una formula lógica compuesto por conjunción de disyunciones:

$$(a_1 \vee b_1) \wedge (a_2 \vee b_2) \wedge \cdots \wedge (a_m \vee b_m),$$

Se te pide asignar el valor lógico de “verdadero” o “falso” tal que la formula lógica compuesta sea “verdadera”.

- Solución trivial

Probar cada asignación a cada variable y evaluar si la expresión es “verdadera”. Complejidad O(2^n)

Problema Ejemplo: 2 SAT

$$L_1 = (x_2 \vee \neg x_1) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_4)$$

$$\begin{cases} x_1 = \text{false} \\ x_2 = \text{false} \\ x_3 = \text{true} \\ x_4 = \text{true} \end{cases}$$

$$L_2 = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$

No existe solución

Dado que entre x2 y !x2 hay un falso, entonces x1 debe ser verdadero

Dado que entre x3 y !x3 hay un falso, entonces x1 debe ser falso

Contradicción, por lo que no existe solucion

Problema Ejemplo: 2 SAT

Disyunción

ϕ	ψ	$\phi \vee \psi$
V	V	V
F	V	V
V	F	V
F	F	F

Conjunción

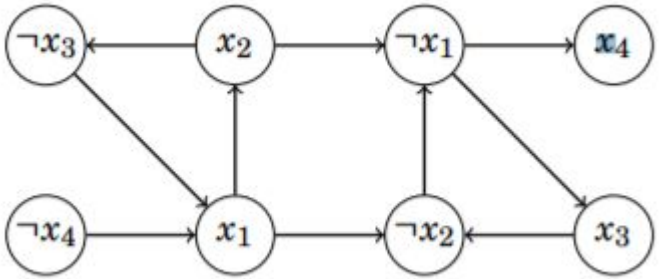
ϕ	ψ	$\phi \wedge \psi$
V	V	V
F	V	F
V	F	F
F	F	F

Cada disyunción por separada debe ser verdadera y para que eso ocurra al menos una de las variables debe ser verdadera

Una disyunción se puede representar como 2 condicionales:
Si a no cumple en ser verdadero, entonces b debe ser verdadero
Si b no cumple en ser verdadero, entonces a debe ser verdadero

Problema Ejemplo: 2 SAT

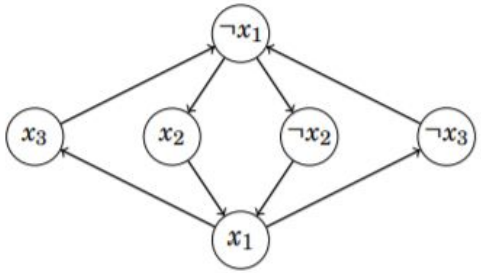
$$L_1 = (x_2 \vee \neg x_1) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_1 \vee x_3) \wedge (\neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_4)$$



Existe un camino entre x1 a -x1.
Pero no existe un camino entre –x1 a x1.

X1 y –X1 no pertenecen a un mismo SCC

$$L_2 = (x_1 \vee x_2) \wedge (x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$



Existe un camino entre x1 a -x1.
Existe un camino entre –x1 a x1.

X1 y –X1 si pertenecen a un mismo SCC.

Para que 2-SAT tenga solución, para cada xi, xi y –xi no debería pertenece a un mismo SCC

Problema Ejemplo: 2 SAT

En el grafo condensado, cada SCC debería tener la misma asignación (verdadero o falso), pero al menos debería existe 1 verdadero y 1 falso.

La única secuencia que cumple con ser verdadera y cumplir con la condiciónal es:

FFFF…VVV

Entonces para reconstruir una solución, podemos usar topological sort inverso e ir asignando las variables en el orden:

$$A = \{\neg x_4\}, B = \{x_1, x_2, \neg x_3\}, C = \{\neg x_1, \neg x_2, x_3\} \text{ and } D = \{x_4\}.$$

X4: Verdadero , X3 : Verdadero, X2: Falso, X1: Falso

¡ Good luck and have fun !