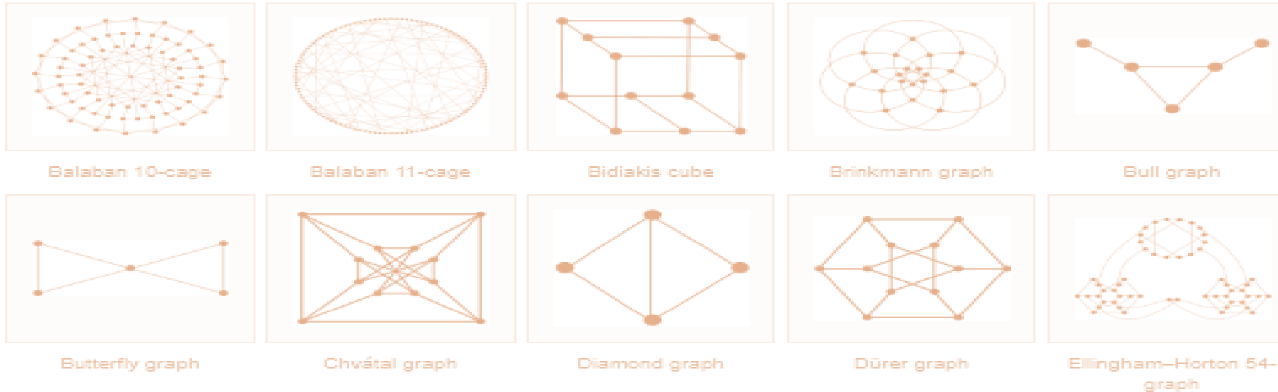
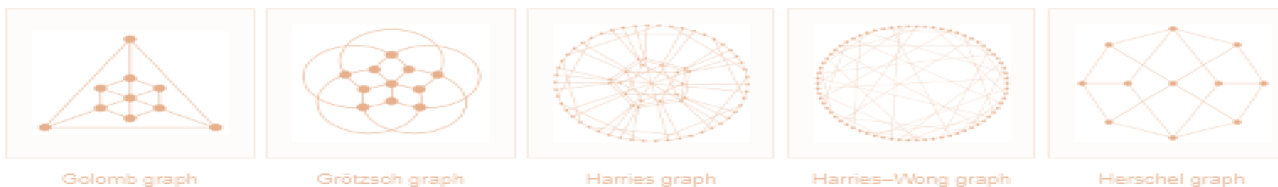
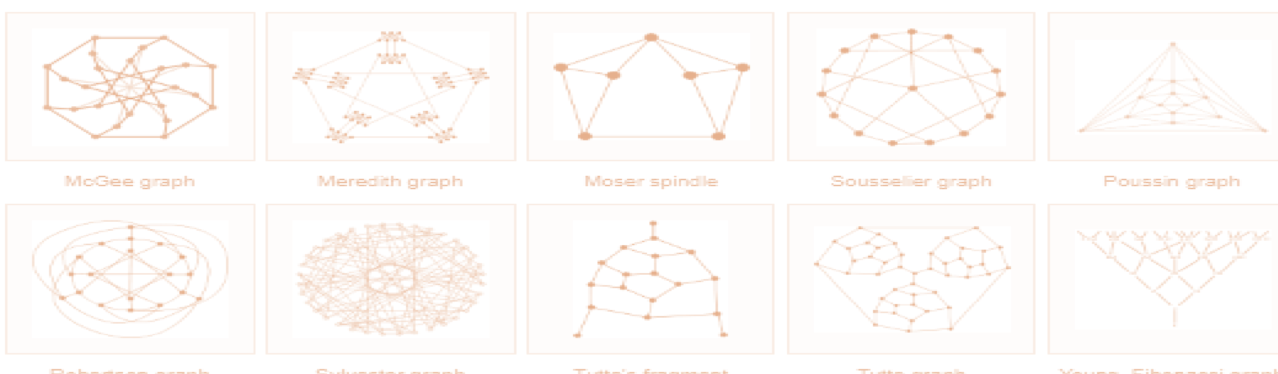


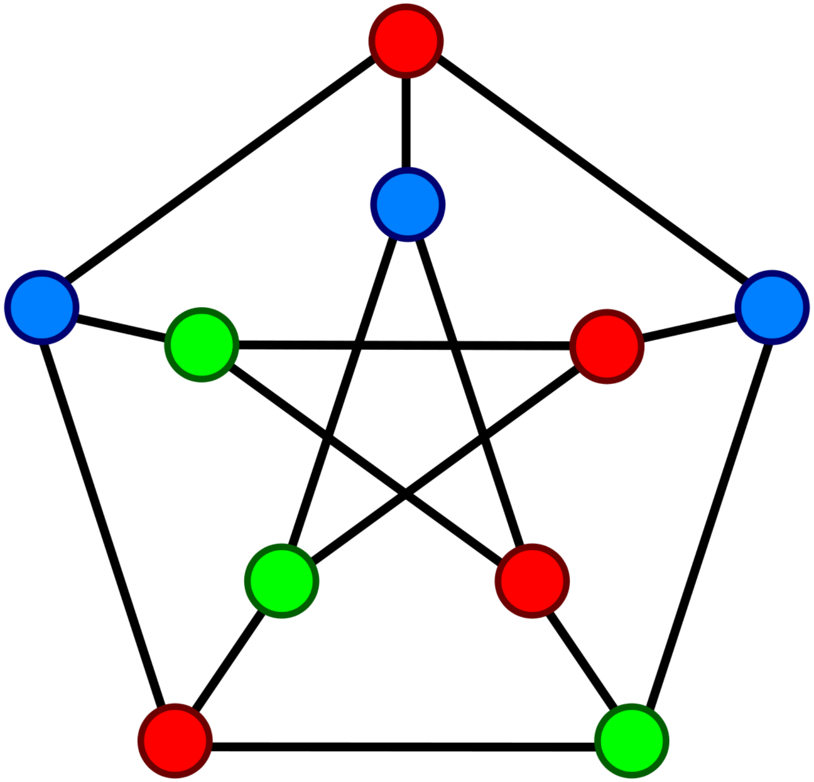
Graph Theory



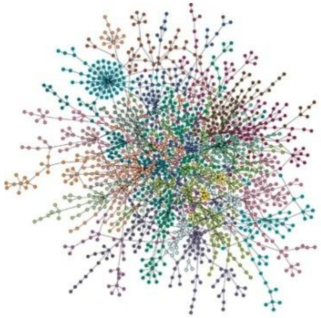
Coloring



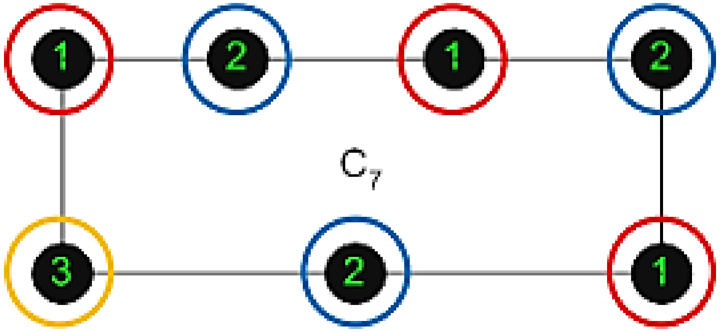
Planarity



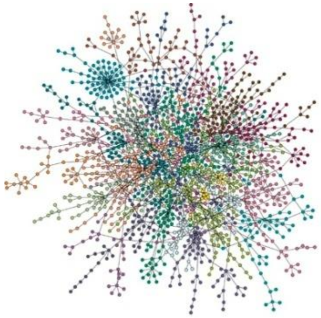
Coloring



- **Graph Coloring**
  - The **coloring** of a graph refers to the process of color rendering on its vertices, so that there are no adjacent vertices of the same color.
  - A vertex set containing vertices of the same color defines a **color class**.
  - Coloring may refer to the color rendering over vertices/edges/regions

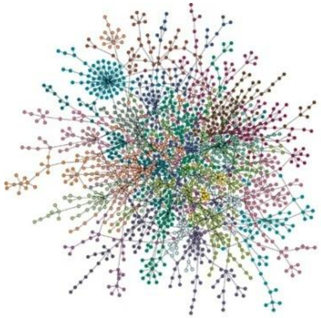


COLORING

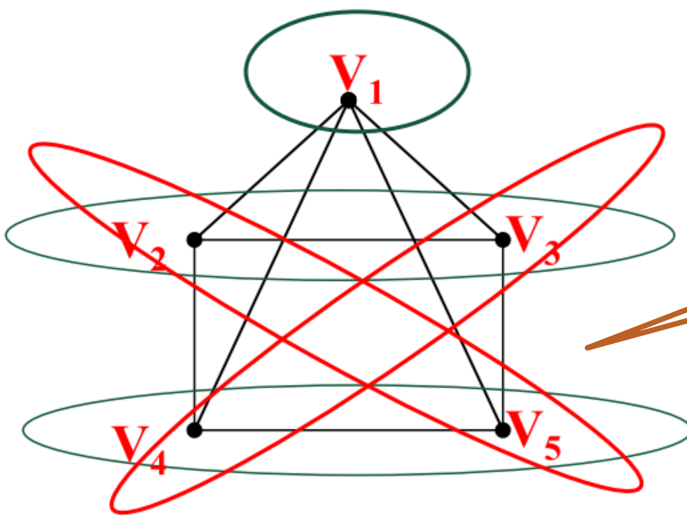


- **Graph Coloring**
  - **K-colorable** graph: vertices can be colored with k colors.
  - **k-chromatic** graph: vertices can be colored with k colors, but not with $k-1$.
  - **Chromatic number**: $\chi\,(G) = k$.

Coloring

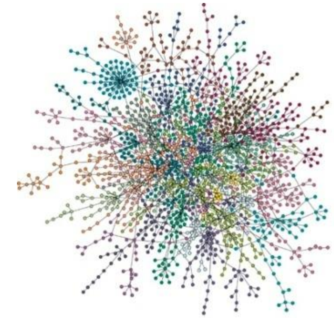


- **Graph Coloring**
  - A **uniquely colorable** graph is the one that has a specific set of color classes, without being able to define a different set.
  - There are special cases for $k = x(G)$, where the color classes are stable.
  - General case for $k > x(G), G$ is colored with many different ways using k colors



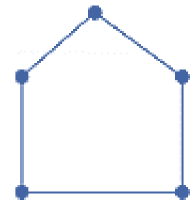
3 color classes:
$\{v_1\}, \{v_2, v_5\}, \{v_3, v_4\}$

Coloring



Graph Coloring

- A graph is critical if it holds: $\chi(H) < x\ (G),\ \forall\ H \subset G$.



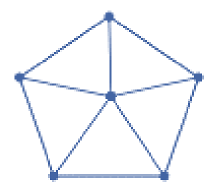
$\chi(G) = 3$
$\chi(H) < 3$

- A graph is k-critical if G is k-chromatic and critical, k $\geq$ 2 if $x(G) = k$ and $\chi(G - v) = k - 1, \forall\ n \in V(G)$.

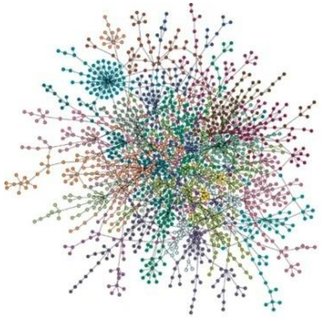
$\chi(W) = 4 \rightarrow$ 4-chromatic and critical

$\rightarrow$ 4-critical

W_6



Coloring



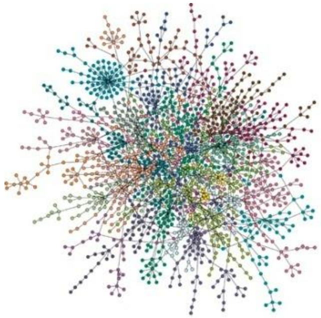
Graph Coloring

- **Theorem 1:**

  If the graph G is k-critical, then it holds: $d(G) \geq\ k-1$.

  - Let us assume that G is k-critical and it holds $d(G) < k-1$.
  - We will color the graph with less than k colors.
  - We select a vertex v: $d(v) = d(G)$
  - Since G is k −colorable $\longrightarrow$ $G - v$ is $(k-1)$ −critical.
  - We color the graph $G - v$ with $k-1$ colors.
  - Let that $V_1, V_2, \dots\ V_{k-1}$ are the corresponding color classes.
  - Since $d(v) = d(G) < k-1$ there should exist a color class V_i such that vertex v not being adjacent to any other vertex of that class.
  - Hence, vertex v can be colored with color i, and hence the graph G can be colored with $k-1$ colors, that is a contradiction.

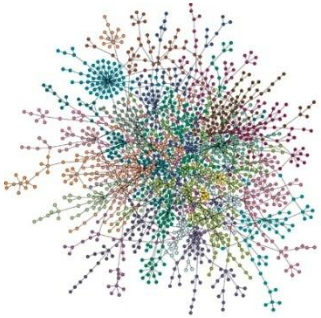
Coloring



Graph Coloring

- A **Perfect graph** is the graph in which it holds that clique number $\omega(H) = x(H)$ color number for each induced subgraph H of graph G.
  - The clique of a graph G is the maximum complete subgraph of G, and the click number $\omega(G)$ denotes the clique-order.
- A **k-edge Colorable** graph is a graph which its edges can be colored with k colors, such that any two edges having a common vertex to have different colors.
- In case where a graph G is k-edge colorable but it is not $k-1$-edge colorable is called **k-edge Chromatic**, and the value of k is called **edge chromatic number**, or, **chromatic index** denoted $x'(G)$.
- If a graph can be colored uniquely on its edges such that the color classes to remain constant, the graph is called **uniquely edge-colorable**.

Coloring



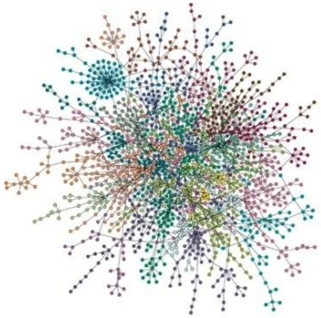
Vertex Coloring

- How many colors are required to color a graph?
  - … each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 2:**

  A graph G of max degree D is $(D + 1)$-colorable.

  - If the graph G has $n = D + 1$ vertices then the truth is obvious.
  - Let that the Theorem holds for $n = k - 1$ vertices.
  - There should be proven that the Theorem holds for G of order k.

Coloring

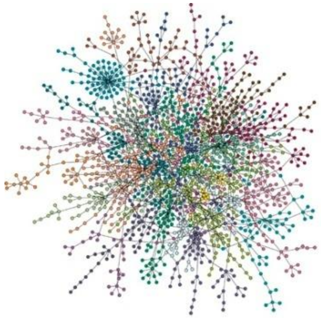


Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 2:**

  A graph G of max degree D is ($D + 1$)-colorable.
  - If from the graph G we delete a vertex of degree at most D, then it remains a subgraph of degree at most D of $n - 1$ vertices, that is $(D + 1) -$ colorable, based on the hypothesis of induction.
  - Based on this subgraph, G can be colored assigning to vertex v a color different from the D neighboring vertices.

COLORING

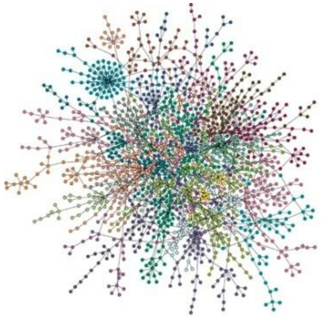


Vertex Coloring

- How many colors are required to color a graph?
  - … each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 3 (Brooks 1941):**

  A simple connected non-complete graph G of max degree $D \geq 3$ is D-colorable.

COLORING

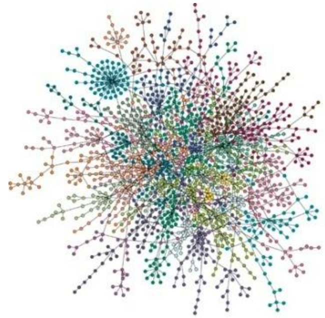


Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 4:**

  Every plane graph is 6-colorable.
  - If the graph G has $n < 7$ vertices then the truth is obvious.
  - Let that the Theorem holds for $n = k - 1$ vertices.
  - There should be proven that the Theorem holds for G of order k.

Coloring

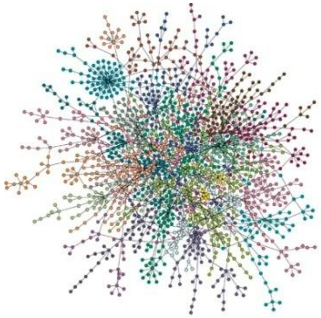


Vertex Coloring

- How many colors are required to color a graph?
  - … each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 4:**

  Every plane graph is 6-colorable.
  - Let us assume that the graph is simple graph.
  - Corollary on plane graphs: "each plane graph has at least one vertex v of $d(v) \leq 5$"
  - From the Corollary, this graph contains at least one vertex of $d(v) \leq 5$

Coloring

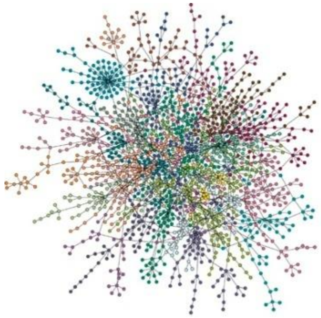


Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 4:**

  Every plane graph is 6-colorable.
  - If we delete from the graph G a vertex v of degree at most 5, then there remains a graph of $n-1$ vertices, which is 6-colorable according to the hypothesis of the induction.
  - Based on this subgraph we can color the graph using a color for v different from the colors of its 5 neighbors.

Coloring

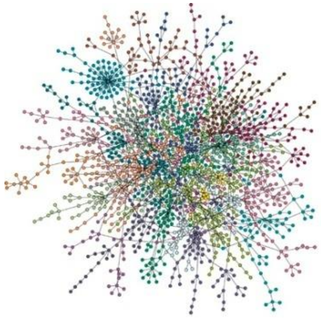


Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 5:**

  Every plane graph is 5-colorable.
  - If the graph G has $n < 6$ vertices then the truth is obvious.
  - Let that the Theorem holds for $n = k - 1$ vertices.
  - There should be proven that the Theorem holds for G of order k.

Coloring

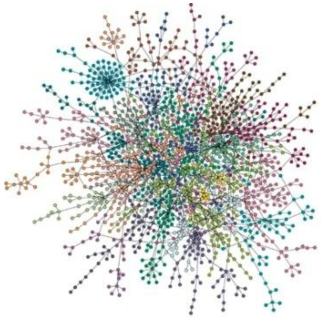


Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 5:**

  Every plane graph is 5-colorable.
  - Let us assume that the graph is simple graph.
  - Corollary on plane graphs: "each plane graph has at least one vertex v of $d(v) \leq 5$"
  - From the Corollary, this graph contains at least one vertex of $d(v) \leq 5$

Coloring

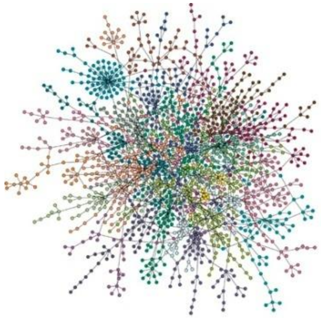


Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 5:**

  Every plane graph is 5-colorable.
  - If we delete from the graph G a vertex v of degree at most 5, then there remains a graph of $n-1$ vertices, which is 5-colorable according to the hypothesis of the induction.
  - If $d(v) < 5$ then the proof is complete.
  - Let that $d(v) = 5$ and the neighboring vertices of v are $v_1, v_2, \dots, v_5$.
  - If two of these vertices have the same color then the proof is complete.

Coloring

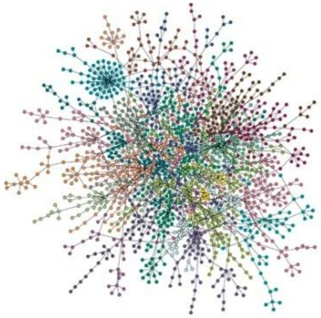


Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 5:**

  Every plane graph is 5-colorable.
  - So, it lefts the case where the five vertices v_i are colored respectively with 5 different colors c_i $(1 \leq \mathrm{i} \leq 5)$.
  - We define a subgraph H_{ij} of G consisted by the vertices of G colored by colors c_i and c_j and the edges of G connecting a vertex of color c_i with a vertex of color c_j.
  - Hence we have two cases:

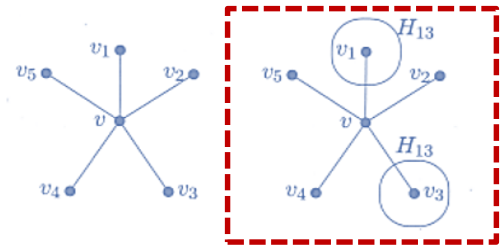
Coloring



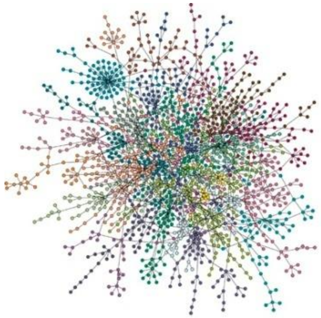
Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 5:**

  Every plane graph is 5-colorable.



  - Case 1:
    - Vertices v_1 and v_3 do not belong to the same component of $H_{1,3}$
    - Hence, a mutual swap of the colors can be performed on the vertices belonging to the component, let the one that belongs vertex v_1.
    - Thus, v_1 can be colored with color c_3 instead of c_1, and hence v_1 can be colored with color c_1

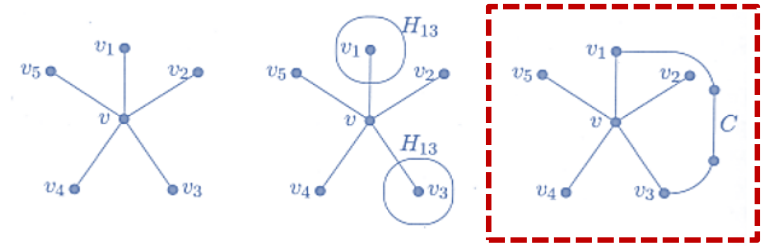


COLORING

Vertex Coloring

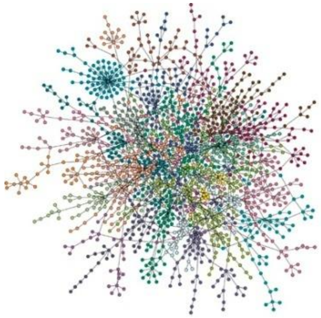
- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 5:**

  Every plane graph is 5-colorable.



  - <u>Case 2</u>:
    - Vertices v_1 and v_3 belong to the same component of $H_{1,3}$
    - In that case there exist a cycle $C = (v, v_1, \dots, v_3, v)$
    - If so, vertex v_2 (v_4 respectively) belongs in the inside (outside respsecively) if cycle C, and hence there does not exist any path from v_2 to v_4 in subgraph $H_{2,4}$.

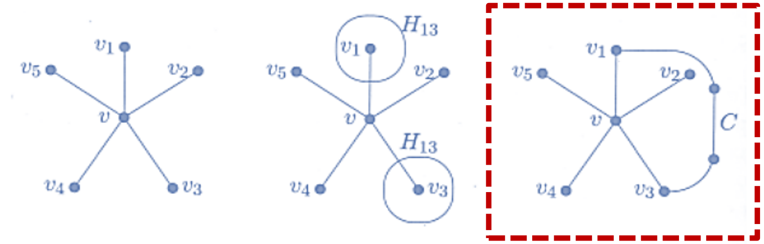
Coloring



Vertex Coloring

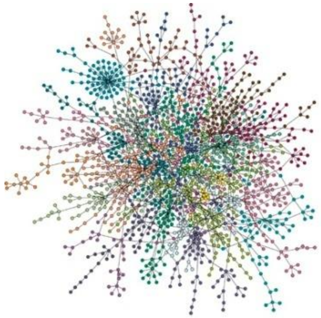
- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 5:**

  Every plane graph is 5-colorable.

  - Case 2:
    - Vertices v_1 and v_3 belong to the same component of $H_{1,3}$
    - In that case there exist a cycle $C = (v, v_1, \dots, v_3, v)$
    - Thus, a mutual swap of the color can be performed on the vertices contained in component $H_{2,4}$, that include vertex v_2.
    - Hence, again, vertex v_2 can be colored with color c_4 instead of c_2, and then vertex v can be colored with color c_2.

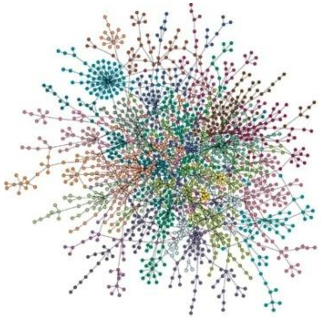
COLORING



- **Vertex Coloring**
  - How many colors are required to color a graph?
    - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
    - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
  - But … may a random graph be colored with less than k colors?
  - **Theorem 6:**

    For a graph G there exist a number k, such that G contains k vertices of degree at least k. Then G can be colored with at most k colors.

COLORING

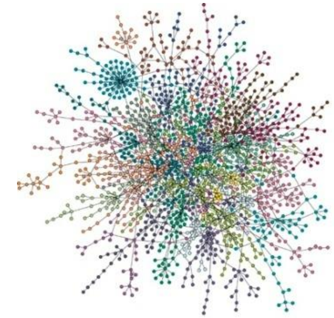


Graph Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 6:**

  For a graph G there exist a number k, such that G contains k vertices of degree at least k. Then G can be colored with at most k colors.
  - Set the degree sequence and below
    set the numbers 0,1,2, …

Coloring



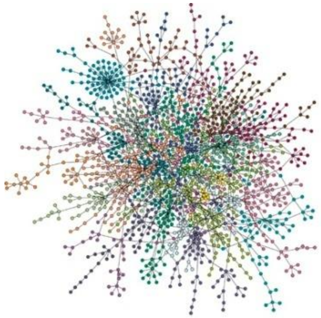
Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 6:**

  For a graph G there exist a number k, such that G contains k vertices of degree at least k. Then G can be colored with at most k colors.
  - Set the degree sequence and below set the numbers 0,1,2, …

d_1	d_2	d_3	…
0	1	2	…

COLORING



Vertex Coloring

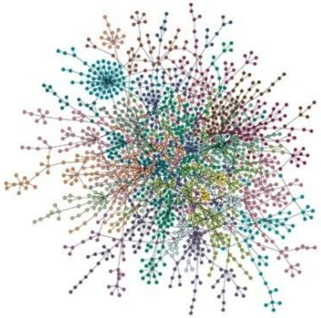
- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where $(k \leq p)$
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 6:**

  For a graph G there exist a number k, such that G contains k vertices of degree at least k. Then G can be colored with at most k colors.
  - Set the degree sequence and below set the numbers 0,1,2, …

Repeat until k is placed above a less degree

d_1	d_2	d_3	…	d_{k+1}
0	1	2	…	k

Coloring



Vertex Coloring

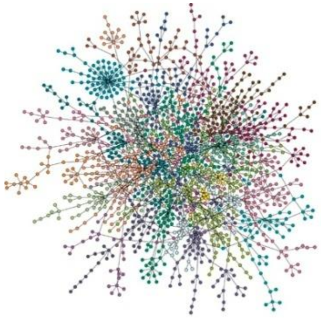
- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where $(k \leq p)$
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 6:**

  For a graph G there exist a number k, such that G contains k vertices of degree at least k. Then G can be colored with at most k colors.
  - Set the degree sequence and below set the numbers 0,1,2, …

…after that, no more colors needed!!!

d_1	d_2	d_3	…	d_{k+1}	…
0	1	2	…	k	…

Coloring



Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 6:**

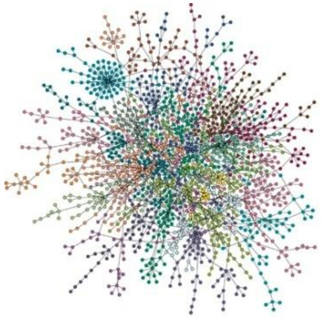
  For a graph G there exist a number k, such that G contains k vertices of degree at least k. Then G can be colored with at most k colors.

  - Set the degree sequence and below set the numbers 0,1,2, …

k is upper bound of $x(G)$!!!

d_1	d_2	d_3	…	d_{k+1}	…
0	1	2	…	k	…

Coloring



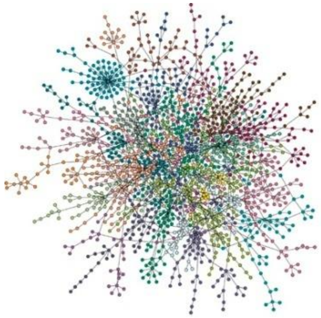
Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 7 (Four Colors Theorem):**

  Every plane graph is 4-colorable:

  - The proof is not include in any educational book!

COLORING



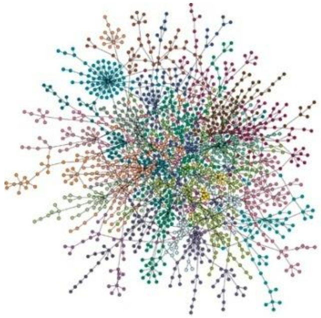
Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where $(k \leq p)$
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- **Theorem 8:**

  Every graph of genus $g \geq 1$ is h-colorable, where:

$$h = \left\lceil \frac{7+\sqrt{1+48g}}{2} \right\rceil$$

COLORING

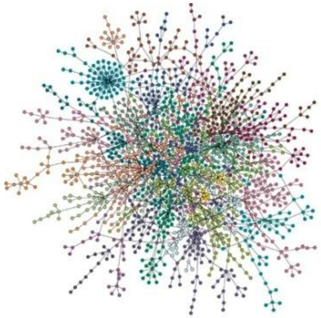


Vertex Coloring

- How many colors are required to color a graph?
  - ... each k-chromatic graph is multi-partite graph of p subsets of vertices, where ($k \leq p$)
  - If the graph is consisted of n vertices, then $x(G) \leq n$, while, if the graph K_r (clique) is contained as subgraph in the graph then $x(G) \geq r$.
- But … may a random graph be colored with less than k colors?
- <u>Chromatic Numbers for most Common Graph Types:</u>

G	K_n	N_n	W_{2n}	W_{2n+1}	P_n	T	C_{2n}	C_{2n+1}	Q_n	$K_{m,n}$
$X(G)$	n	1	4	3	2	2	2	3	n	2

Coloring



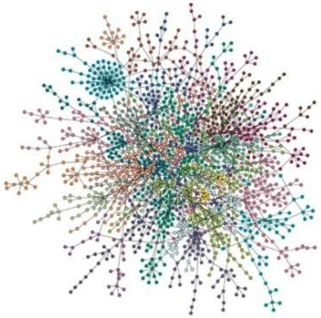
- **"Map" Coloring**
  - **Theorem 9:**

Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

$(\Rightarrow)$

- Let that G is a simple plane graph. Then it is impled that there exists dual of G, $\tilde{G}$ that is map.
- If G is k-colorable, then since every region of $\tilde{G}$ contains only one vertex of G, the regions of $\tilde{G}$ can be colored such that each region to inherit the colors of the corresponding vertex of G.
- Since any two adjacent vertices of G are colored wit different colors, it is implied that any two adjacent regions of $\tilde{G}$ are colored with different colors.

Coloring



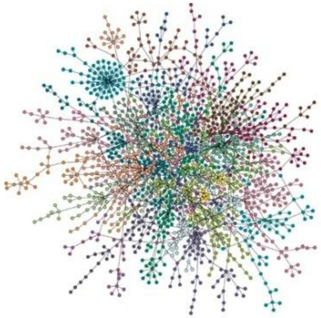
- **“Map” Coloring**
  - **Theorem 9:**

    Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

    ($\Leftarrow$)

    - Let that $\tilde{G}$ is k-colorable, with respect to its regions.
    - Since since every vertex of G is contained in a region of $\tilde{G}$, it is implied that the vertices if G can be colored with k colors, assigning to each vertex the color of the regions it is contained.
    - Since any two adjacent regions of $\tilde{G}$ are colored with different colors it is implied that any two adjacent vertices of G are colored with different colors.

COLORING



- **"Map" Coloring**
  - **Theorem 9:**

    Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

  - **Corollary:**

    The Four Colors Theorem for the coloring of regions of maps equals the coloring (four color coloring) of the vertices of simple plane graphs.

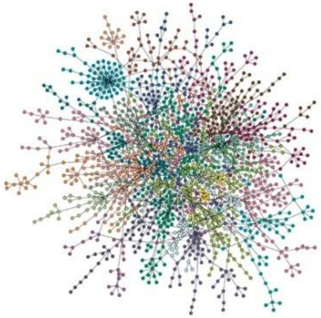
  - **Theorem 10:**

    A map G is 2-colorable with respect to its regions iff G is Eulerian

    $(\Rightarrow)$

    - Since G is 2-colorable with respect to its edges it is implied that each vertex v is surrounded by even number of regions.
    - Thus, each vertex has even degree → The graph is Eulerian.

Coloring



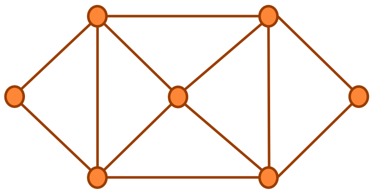
- **"Map" Coloring**
  - **Theorem 9:**

    Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.
  - **Corollary:**

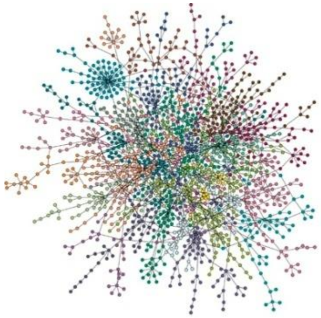
    The Four Colors Theorem for the coloring of regions of maps equals the coloring (four color coloring) of the vertices of simple plane graphs.
  - **Theorem 10:**

    A map G is 2-colorable with respect to its regions iff G is Eulerian

    $(\Leftarrow)$
    - Coloring the regions of G.



Coloring



“Map” Coloring

- **Theorem 9:**

  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

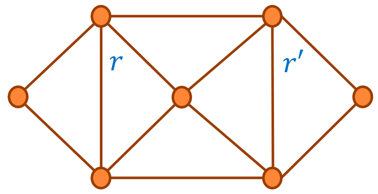
- **Corollary:**

  The Four Colors Theorem for the coloring of regions of maps equals the coloring (four color coloring) of the vertices of simple plane graphs.

- **Theorem 10:**

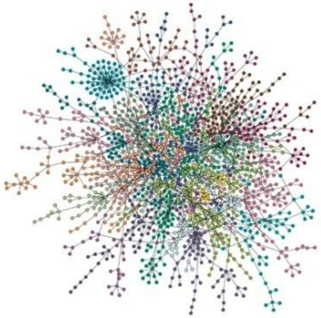
  A map G is 2-colorable with respect to its regions iff G is Eulerian

  $(\Leftarrow)$

  - Coloring the regions of G.
  - Arbitrarily select a region r which we color as “blue”

Coloring



“Map” Coloring

- **Theorem 9:**

  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

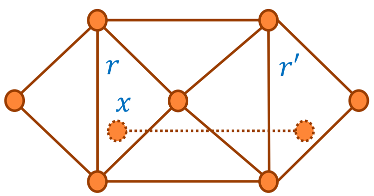
- **Corollary:**

  The Four Colors Theorem for the coloring of regions of maps equals the coloring (four color coloring) of the vertices of simple plane graphs.

- **Theorem 10:**

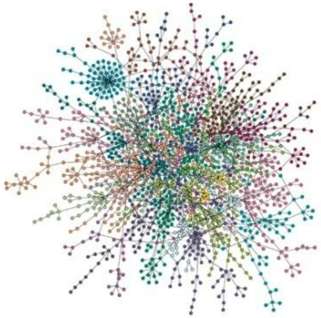
  A map G is 2-colorable with respect to its regions iff G is Eulerian

  $(\Leftarrow)$



  - Coloring the regions of G.
  - Arbitrarily select a region r which we color as “blue”
  - Draw a Jordan curve from a point x in r to finishing in r', crossing no edges in the middle.

Coloring



“Map” Coloring

- **Theorem 9:**

  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

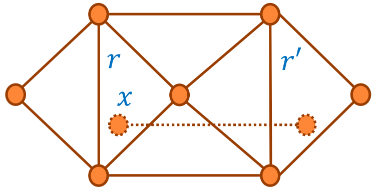
- **Corollary:**

  The Four Colors Theorem for the coloring of regions of maps equals the coloring (four color coloring) of the vertices of simple plane graphs.

- **Theorem 10:**

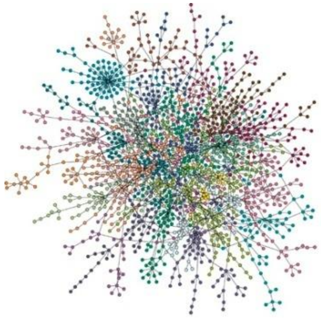
  A map G is 2-colorable with respect to its regions iff G is Eulerian

  $(\Leftarrow)$

  - Coloring the regions of G.
  - Arbitrarily select a region r which we color as “blue”
  - If this curve crosses odd number of edges of G, then the region r' is colored as “green”, or as “blue” otherwise.

Coloring



"Map" Coloring

- **Theorem 9:**

  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

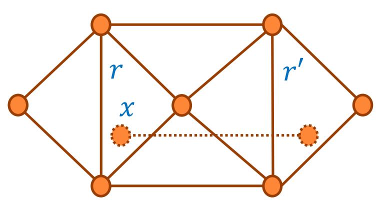
- **Corollary:**

  The Four Colors Theorem for the coloring of regions of maps equals the coloring (four color coloring) of the vertices of simple plane graphs.

- **Theorem 10:**

  A map G is 2-colorable with respect to its regions iff G is Eulerian

  $(\Leftarrow)$



  - Coloring the regions of G.
  - Arbitrarily select a region r which we color as "blue"
  - We assume a closed Jordan curve and proving that it crosses even number of edges.

COLORING

"Map" Coloring

- **Theorem 9:**

  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

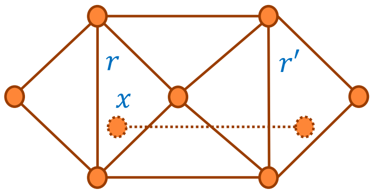
- **Corollary:**

  The Four Colors Theorem for the coloring of regions of maps equals the coloring (four color coloring) of the vertices of simple plane graphs.

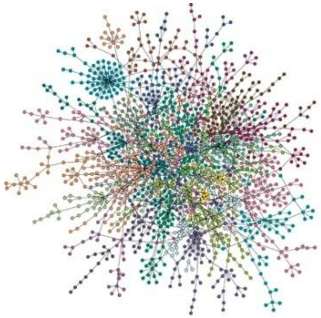
- **Theorem 10:**

  A map G is 2-colorable with respect to its regions iff G is Eulerian

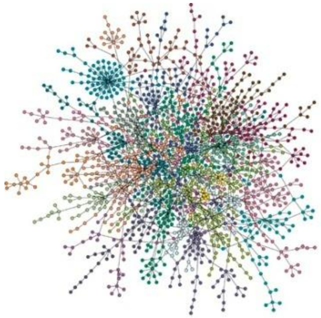
  ($\Leftarrow$)



  - Coloring the regions of G.
  - Arbitrarily select a region r which we color as "blue"
  - The proof can be done inductively with respect to the number of vertices contained in the interior of this curve, taking into account that the degree of each vertex is even.



Coloring



- **“Map” Coloring**
  - **Theorem 9:**

    Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

  - **Theorem 10:**

    A map G is 2-colorable with respect to its regions iff G is Eulerian

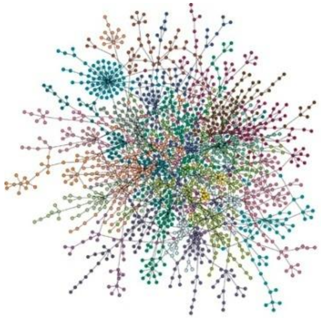
  - **Theorem 11:**

    A regular of degree 3 graph G is 3-colorable with respect to its regions iff each region of G is surrounded by even number of edges.

    $(\Rightarrow)$

    - Given a region r of G, the neighboring regions should be colored with 2 alternating colors.
    - Thus, in order to adequate 3 colors, each region should be surrounded by even number of regions.

Coloring



“Map” Coloring

- **Theorem 9:**

  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

- **Theorem 10:**

  A map G is 2-colorable with respect to its regions iff G is Eulerian

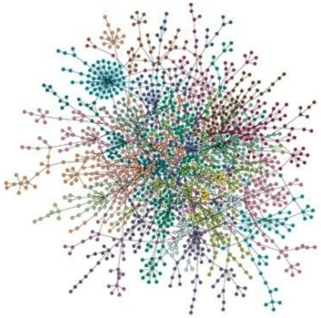
- **Theorem 11:**

  A regular of degree 3 graph G is 3-colorable with respect to its regions iff each region of G is surrounded by even number of edges.

  $(\Leftarrow)$

  - The inverse direction will be proven regarding the geometrical dual of the graph.
  - If the connected graph G is plane and simple, while each region is surrounded by triangles and each vertex has even degree, then it is 3-colorable.

COLORING



“Map” Coloring

- **Theorem 9:**

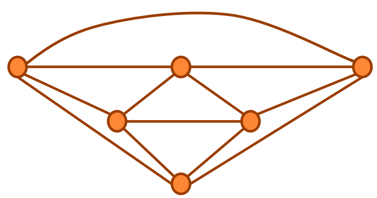
  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

- **Theorem 10:**

  A map G is 2-colorable with respect to its regions iff G is Eulerian

- **Theorem 11:**

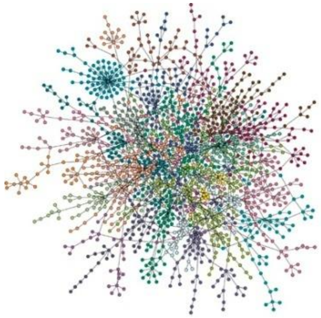
  A regular of degree 3 graph G is 3-colorable with respect to its regions iff each region of G is surrounded by even number of edges.



  $(\Leftarrow)$

  - The three colors are denoted as “a, b, c”
  - If G is Eulerian, the regions of G can be colored with two colors, “blue” and “green”.
  - The demanded 3-coloring, can be achieved as:

COLORING



“Map” Coloring

- **Theorem 9:**

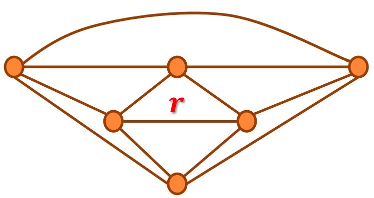
  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

- **Theorem 10:**

  A map G is 2-colorable with respect to its regions iff G is Eulerian

- **Theorem 11:**

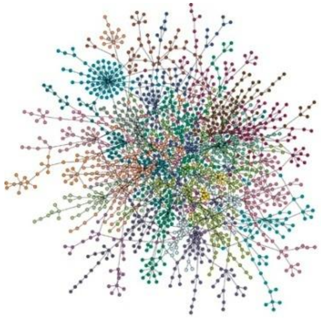
  A regular of degree 3 graph G is 3-colorable with respect to its regions iff each region of G is surrounded by even number of edges.

  $(\Leftarrow)$

  - The three colors are denoted as “a, b, c”
  - If G is Eulerian, the regions of G can be colored with two colors, “blue” and “green”.
  - The demanded 3-coloring, can be achieved as:
    1. Select a region r (let “blue”).

COLORING



“Map” Coloring

- **Theorem 9:**

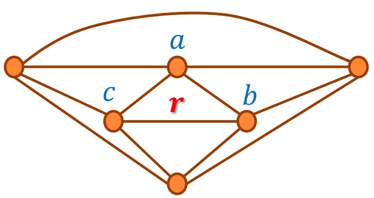
  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

- **Theorem 10:**

  A map G is 2-colorable with respect to its regions iff G is Eulerian

- **Theorem 11:**

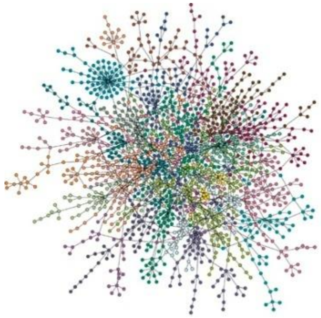
  A regular of degree 3 graph G is 3-colorable with respect to its regions iff each region of G is surrounded by even number of edges.



  $(\Leftarrow)$

  - The three colors are denoted as “a, b, c”
  - If G is Eulerian, the regions of G can be colored with two colors, “blue” and “green”.
  - The demanded 3-coloring, can be achieved as:
    2. Its vertices are colored by 3 colors (clockwise a, b, c).

Coloring



“Map” Coloring

- **Theorem 9:**

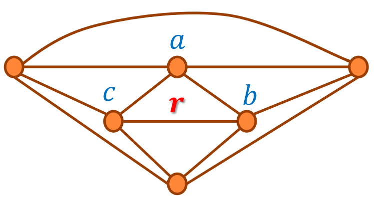
  Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

- **Theorem 10:**

  A map G is 2-colorable with respect to its regions iff G is Eulerian

- **Theorem 11:**

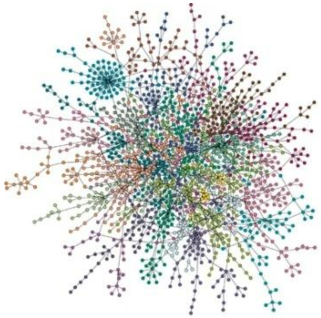
  A regular of degree 3 graph G is 3-colorable with respect to its regions iff each region of G is surrounded by even number of edges.

  $(\Leftarrow)$

  - The three colors are denoted as “a, b, c”
  - If G is Eulerian, the regions of G can be colored with two colors, “blue” and “green”.
  - The demanded 3-coloring, can be achieved as:
    3. Select another region (let “blue”).

COLORING



- **“Map” Coloring**
  - **Theorem 9:**

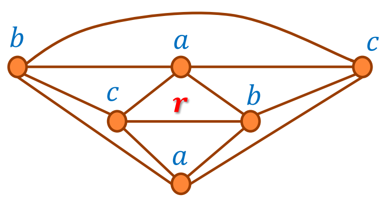
    Let a simple connected graph G and $\tilde{G}$ its dual. G is k-colorable if and only if $\tilde{G}$ is k-colorable with respect to its regions.

  - **Theorem 10:**

    A map G is 2-colorable with respect to its regions iff G is Eulerian

  - **Theorem 11:**

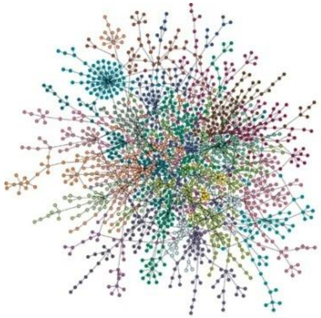
    A regular of degree 3 graph G is 3-colorable with respect to its regions iff each region of G is surrounded by even number of edges.



$(\Leftarrow)$

- The three colors are denoted as “a, b, c”
- If G is Eulerian, the regions of G can be colored with two colors, “blue” and “green”.
- The demanded 3-coloring, can be achieved as:
  4. Repeat the procedure…

Coloring



Edge Coloring

- **Theorem 12 (Vizing 1964):**

  Let a simple graph G of maximum degree $\mathrm{D}(G)$, then it holds:

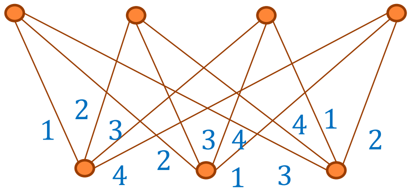
  $$D(G) \leq x'(G) \leq D(G) + 1$$

- **Theorem 13:**

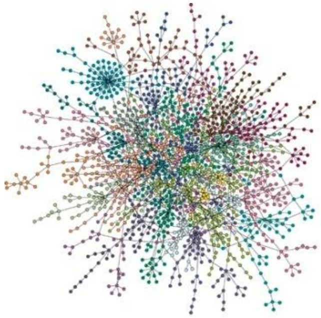
  For a complete bipartite graph it holds:

  $$x'(K_{m,n}) = \max(m, n) = D(K_{m,n})$$

  - Without loss of generality, let that $m \geq n$, and that the m vertices are located on a straight line, while the n vertices are located below over another straight line.
  - Coloring is achieved by coloring sequentially (clockwise) the edges adjacent to the n vertices with colors $\{\mathbf{1}, \mathbf{2}, \dots, \boldsymbol{m}\}, \{\mathbf{2}, \mathbf{3}, \dots, \boldsymbol{m}, \mathbf{1}\}, \dots, \{\boldsymbol{n}, \dots, \boldsymbol{m}, \mathbf{1}, \dots, \boldsymbol{n} - \mathbf{1}\}$.
  - The Theorem holds also for non complete graphs.



Coloring



Edge Coloring

- **Theorem 12 (Vizing 1964):**

  Let a simple graph G of maximum degree $\mathrm{D}(G)$, then it holds:

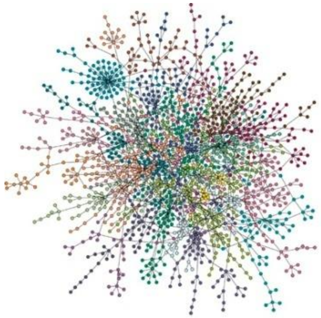
$$D(G) \leq x'(G) \leq D(G) + 1$$

- **Theorem 14:**

  For a complete graph K_n it holds:

$$x'(K_n) = \begin{cases} n, & if\ n\ is\ odd \\ n-1, & if\ n\ is\ even \end{cases}$$

Coloring



Edge Coloring

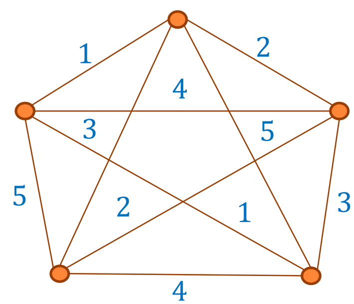
- **Theorem 12 (Vizing 1964):**

  Let a simple graph G of maximum degree $\mathrm{D}(G)$, then it holds:

  $$D(G) \leq x'(G) \leq D(G) + 1$$

- **Theorem 14:**

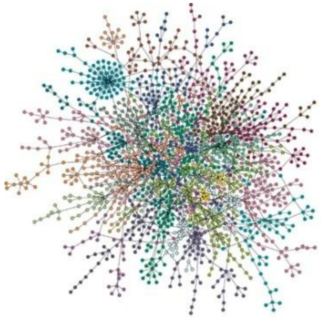
  For a complete graph K_n it holds:



$$x'(K_n) = \begin{cases} n, & if\ n\ is\ odd \\ n-1, & if\ n\ is\ even \end{cases}$$

- The graph can be colored with n colors placing the vertices in shape of regular n −polygon, and coloring clockwise the edges in the perimeter.
- The rest of the edges are colored with the same color that is colored the parallel to the perimeter.
- The graph K_n is k-edge chromatic because the maximum number of edges of the same color is $(n-1)/2$

COLORING



Edge Coloring

- **Theorem 12 (Vizing 1964):**

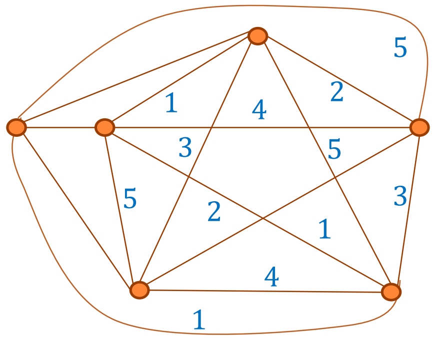
  Let a simple graph G of maximum degree $\mathrm{D}(G)$, then it holds:

  $$D(G) \leq x'(G) \leq D(G) + 1$$

- **Theorem 14:**

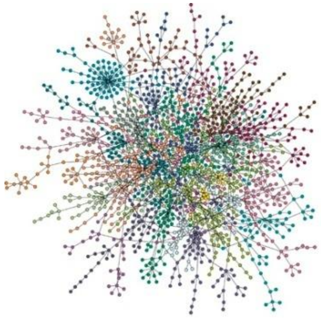
  For a complete graph K_n it holds:

  $$x'(K_n) = \begin{cases} n, & \mathit{if\ n\ is\ odd} \\ n-1, & \mathit{if\ n\ is\ even} \end{cases}$$



  - The graph K_n can be considered as the sum of K_{n-1} and K_1.
  - If the edges of K_{n-1} have been colored according to the previous approach, then from each vertex a color is missing.
  - For a each vertex will be missing a different color.
  - Thus, the edges adjacent to K_1, will be colored with these colors.

Coloring



Edge Coloring

- **Theorem 12 (Vizing 1964):**

  Let a simple graph G of maximum degree $\mathrm{D}(G)$, then it holds:

  $$D(G) \le x'(G) \le D(G) + 1$$

- **Theorem 14:**

  For a complete graph K_n it holds:

  $$x'(K_n) = \begin{cases} n, & if\ n\ is\ odd \\ n-1, & if\ n\ is\ even \end{cases}$$

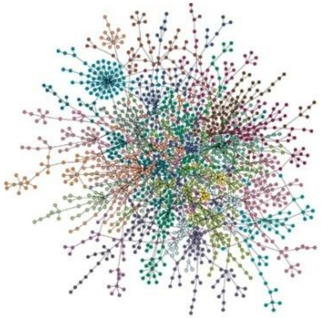
- **Theorem 15 (Vizing 1965):**

  If G is a simple multigraph then it holds:

  $$D(G) \le x'(G) \le D(G) + m(G)$$

  Maximum multiplicity $m(G)$ is the maximum number of edges joining any pair of vertices in a multigraph.

COLORING



- **Edge Coloring**
  - **Theorem 12 (Vizing 1964):**

    Let a simple graph G of maximum degree $\mathrm{D}(G)$, then it holds:

    $$D(G) \leq x'(G) \leq D(G) + 1$$

  - **Theorem 14:**

    For a complete graph K_n it holds:

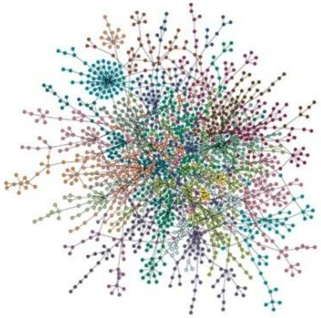
    $$x'(K_n) = \begin{cases} n, & if\ n\ is\ odd \\ n-1, & if\ n\ is\ even \end{cases}$$

  - **Theorem 16 (Tait 1980):**

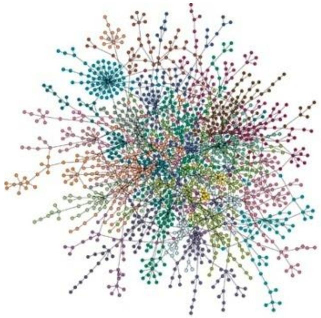
    A map G is 4-colorable w.r.t regions iff it is 3-edge colorable.

Coloring

- **Chromatic Polynomial**
  - Is it possible to color vertices/edges/regions with k colors?



Coloring



Chromatic Polynomial

- Is it possible to color vertices/edges/regions with k colors? **Qualitative**
- By how many ways a given graph can be colored using k colors? **Quantitive**
- Two graphs are considered to be differently colored if at least one vertex has different color.
- Birkhoff (1912) → Four Color Thoerem.
- **Chromatic Polynomial** (**Chromatic Function**) refers to the number of ways a given graph G can be colored with k colors, and is denoted as $\boldsymbol{P_G(k)}$.

$$P_{Nn}(k) = k^n$$
$$P_T(k) = k(k-1)^n$$
$$P_{Kn}(k) = k(k-1)\dots(k-n+1)$$
$$P_G(k) = 0, \text{ if } k < \chi(G)$$
$$P_G(k) > 0, \text{ if } k \geq \chi(G)$$
$$P_G(k) > 0, \text{ if } G \text{ is simple plane graph}$$

Coloring

Chromatic Polynomial

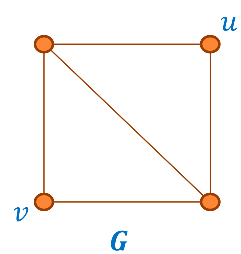
- **Theorem 17:**

  Let that v, w are non adjacent vertices of a simple graph G.
  If $G_1 = G + (v, w)$ and $G_2 = G/(v, w)$, then it holds:

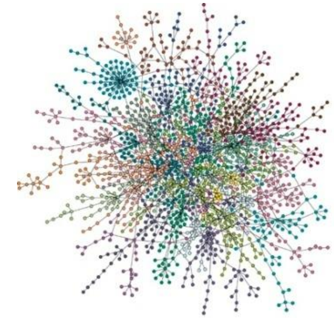
$$P_G(k) = P_{G_1}(k) + P_{G_1}(k)$$

  - The graph G_1 is constructed joining the vertices v and u, while G_2 is constructed merging the vertices v and u considering the induced graph of the constructed multigraph.

  - In each coloring of graph G, the vertices v and u may or may not have the same color.



COLORING



Chromatic Polynomial

- **Theorem 17:**

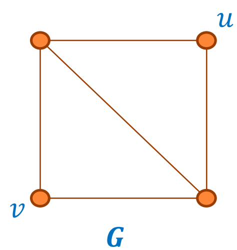
  Let that v, w are non adjacent vertices of a simple graph G.
  If $G_1 = G + (v, w)$ and $G_2 = G/(v, w)$, then it holds:

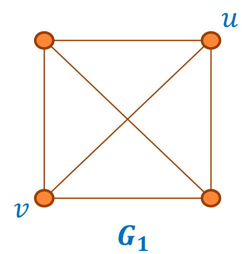
$$P_G(k) = P_{G_1}(k) + P_{G_1}(k)$$

- The graph G_1 is constructed joining the vertices v and u, while G_2 is constructed merging the vertices v and u considering the induced graph of the constructed multigraph.

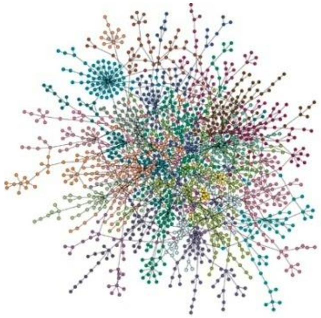
- In each coloring of graph G, the vertices v and u may or may not have the same color.

  - If v and u have different colors, the number of different coloring is retained (i.e., $P_{G_1}(k)$) if the vertices are joined.





COLORING



Chromatic Polynomial

- **Theorem 17:**

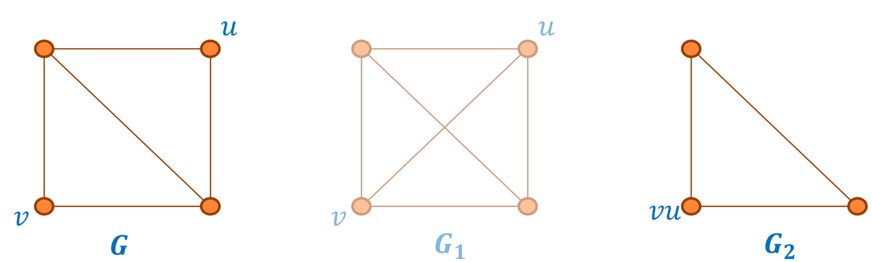
  Let that v, w are non adjacent vertices of a simple graph G.
  If $G_1 = G + (v, w)$ and $G_2 = G/(v, w)$, then it holds:

$$P_G(k) = P_{G_1}(k) + P_{G_1}(k)$$

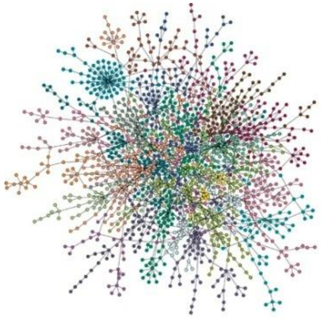
  - The graph G_1 is constructed joining the vertices v and u, while G_2 is constructed merging the vertices v and u considering the induced graph of the constructed multigraph.

  - In each coloring of graph G, the vertices v and u may or may not have the same color.

    - If v and u have same color, the number of different coloring is retained (i.e., $P_{G_2}(k)$) if the vertices are merged.



Coloring



Chromatic Polynomial

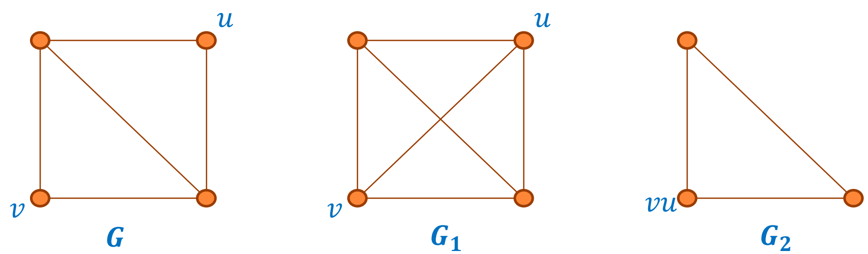
- **Theorem 17:**

  Let that v, w are non adjacent vertices of a simple graph G.
  If $G_1 = G + (v, w)$ and $G_2 = G/(v, w)$, then it holds:

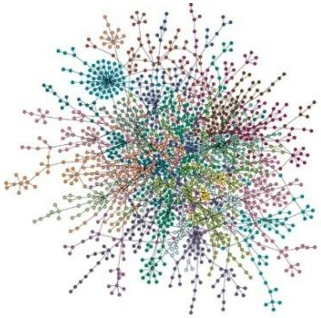
$$P_G(k) = P_{G_1}(k) + P_{G_1}(k)$$

- The graph G_1 is constructed joining the vertices v and u, while G_2 is constructed merging the vertices v and u considering the induced graph of the constructed multigraph.

- In each coloring of graph G, the vertices v and u may or may not have the same color.
  - If v and u have same color, the number of different coloring is retained (i.e., $P_{G\,2}(k)$) if the vertices are merged.



COLORING



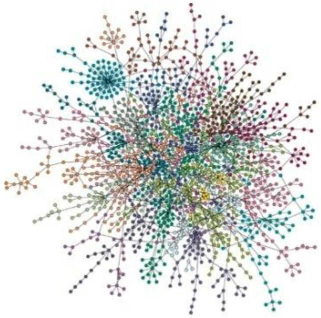
- **Chromatic Polynomial**
  - **Theorem 17:**

    Let that v, w are non adjacent vertices of a simple graph G.
    If $G_1 = G + (v, w)$ and $G_2 = G/(v, w)$, then it holds:

    $$P_G(k) = P_{G_1}(k) + P_{G_1}(k)$$

  - The relations can also be written as: $P_{G_1}(k) = P_G(k) - P_{G_1}(k)$
    - For graphs with many edges $P_G(k) = P_{G_1}(k) + P_{G_1}(k)$ is preferred.
    - For graphs with less edges is preferred $P_{G_1}(k) = P_G(k) - P_{G_1}(k)$

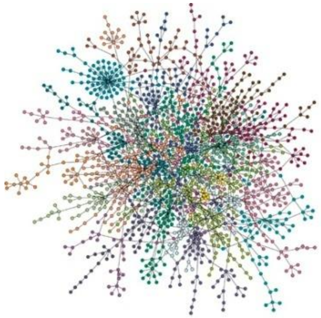
Coloring



- **Chromatic Polynomial**
  - **Corollary (Birkhoff 1912):**

    The chromatic function of a simple graph G of order n is a polynomial w.r.t. k of order n. This polynomial has integer factors of alternating sign, as greater term the K^n and as stable term the 0.

Coloring



Chromatic Polynomial

- **Corollary (Birkhoff 1912):**

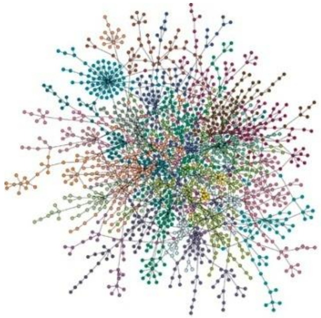
The chromatic function of a simple graph G of order n is a polynomial w.r.t. k of order n. This polynomial has integer factors of alternating sign, as greater term the K^n and as stable term the 0.

- Inductively on the number of edges of graph.
- If G is null, then it holds $P_{N_n}(k) = k^n$.
- Let as assume that the proposition holds for $m-1$ edges.
- We shall prove that it holds when G has m edges:
  - Let e a random edge of G
  - The graphs $G' = G - e$ and $G'' = G \cdot e$ have $k-1$ edges and according to assumption it holds:

$$P_{G'}(k) = \sum_{i=1}^{n-1}(-1)^{n-i}\ r_i k^i + k^n, \text{ and}$$

$$P_{G''}(k) = \sum_{i=1}^{n-2}(-1)^{n-i-1}\ s_i k^i + k^{n-1}$$

Coloring



Chromatic Polynomial

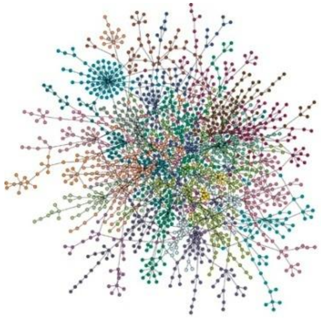
- **Corollary (Birkhoff 1912):**

  The chromatic function of a simple graph G of order n is a polynomial w.r.t. k of order n. This polynomial has integer factors of alternating sign, as greater term the K^n and as stable term the 0.

  - Inductively on the number of edges of graph.
  - If G is null, then it holds $P_{N_n}(k) = k^n$.
  - Let as assume that the proposition holds for $m-1$ edges.
  - We shall prove that it holds when G has m edges:
    - Let e a random edge of G
    - The graphs $G' = G - e$ and $G'' = G \cdot e$ have $k-1$ edges and according to assumption it holds:

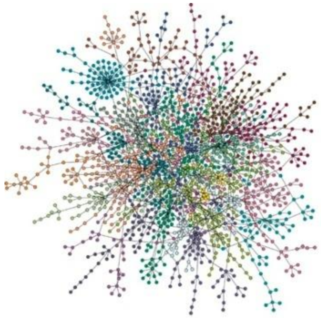
$$\boldsymbol{P_G(k) = P_{G'}(k) + P_{G''}(k) = \sum_{i=1}^{n-2}(-1)^{n-i}\,(r_i + s_i)k^i - (r_{n-1} + 1)k^{n-1} + k^n}$$

Coloring



- **Approximate Coloring Algorithms**
  - **The Problem:**
    - Finding the chromatic number of a random graph is an intractable problem.
    - For small graphs exhaustive search may find this number, but for large graphs any try is ineffective.
    - Therefore, there have been proposed approximate algorithms.
      1. Serial Graph Coloring
      2. Largest First
      3. Smallest Last (Matula – Marble – Isaacson)
      4. Color Degree Method (Brelaz)

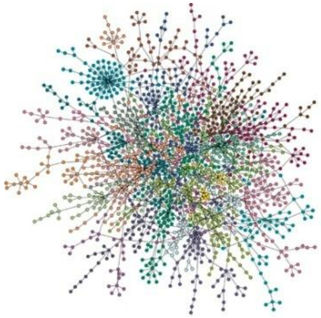
Coloring



Approximate Coloring Algorithms

- **The Problem:**
  - Finding the chromatic number of a random graph is an intractable problem.
  - For small graphs exhaustive search may find this number, but for large graphs any try is ineffective.
  - Therefore, there have been proposed approximate algorithms.
    1. **Serial Graph Coloring**
    2. Largest First
    3. Smallest Last (Matula – Marble – Isaacson)
    4. Color Degree Method (Brelaz)

Coloring



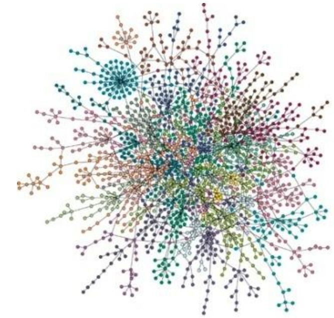
- **Approximate Coloring Algorithms**
  - **Serial Graph Coloring**

Algorithm Serial Coloring

Input: An arbitrarily enumeration on the vertices of $G(V, E)$
Output: A coloring of the graph.

1) $i \leftarrow 1$
2) $c \leftarrow 1$
3) Create a list L_i with the adjacent colors of vertex v_i in descending order.
4) While $c \in L_i$
   $c \leftarrow c + 1$
5) Color vertex v_i with color c
6) If $(i < n)$
   $i \leftarrow i + 1$
   Goto Step 2
7) Else
   exit()

Coloring



- **Approximate Coloring Algorithms**
  - **Serial Graph Coloring**

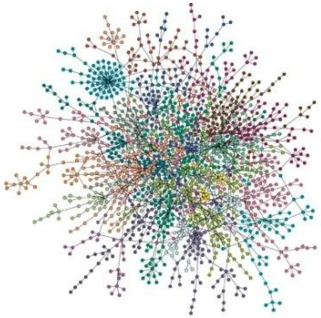
Algorithm Serial Coloring

Input: An arbitrarily enumeration on the vertices of $G(V,E)$
Output: A coloring of the graph.

1) $i \leftarrow 1$
2) $c \leftarrow 1$
3) Create a list L_i with the adjacent colors of vertex v_i in **descending order**.
4) While $c \in L_i$
   $c \leftarrow c + 1$
5) Color vertex v_i with color c
6) If $(i < n)$
   $i \leftarrow i + 1$
   Goto Step 2
7) Else
   exit()

BubbleSort
$\binom{d(v_i)}{2}$ **Comparisons - worst case**

Coloring



Approximate Coloring Algorithms

- **Serial Graph Coloring**

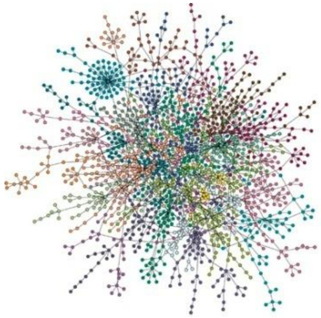
<u>Algorithm Serial Coloring</u>

Input: An arbitrarily enumeration on the vertices of $G(V, E)$
Output: A coloring of the graph.

1) $i \leftarrow 1$
2) $c \leftarrow 1$
3) Create a list L_i with the adjacent colors of vertex v_i in descending order.
4) While $\boldsymbol{c \in L_i}$
   $c \leftarrow c + 1$
5) Color vertex v_i with color c
6) If $(i < n)$
   $i \leftarrow i + 1$
   Goto Step 2
7) Else
   exit()

Binary Search
$\boldsymbol{(d(v_i) + 1) log(d(v_i))}$
comparisons - worst case

Coloring



Approximate Coloring Algorithms

- **Serial Graph Coloring**

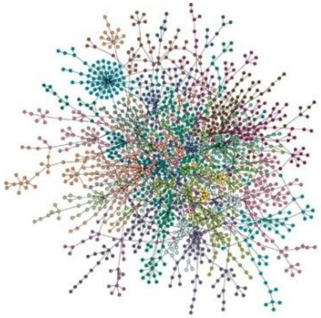
  Algorithm Serial Coloring

  Input: An arbitrarily enumeration on the vertices of $G(V,E)$
  Output: A coloring of the graph.

  1) $i \leftarrow 1$
  2) $c \leftarrow 1$
  3) Create a list L_i with the adjacent colors of vertex v_i in descending order.
  4) While $c \in L_i$
     $c \leftarrow c + 1$
  5) Color vertex v_i with color c
  6) If $(i < n)$
     $i \leftarrow i + 1$
     Goto Step 2
  7) Else
     exit()

$$\boxed{\sum_{i=1}^{n} \binom{d(v_i)}{2} + (d(v_i) + 1)\log(d(v_i))}$$

Coloring



Approximate Coloring Algorithms

- **Serial Graph Coloring**

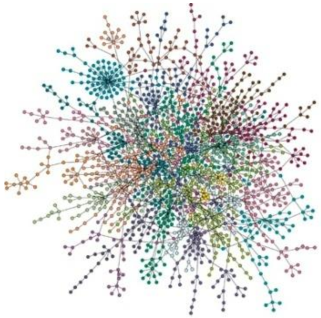
  Algorithm Serial Coloring

  Input: An arbitrarily enumeration on the vertices of $G(V,E)$
  Output: A coloring of the graph.

1) $i \leftarrow 1$
2) $c \leftarrow 1$
3) Create a list L_i with the adjacent colors of vertex v_i in descending order.
4) While $c \in L_i$
   $c \leftarrow c + 1$
5) Color vertex v_i with color c
6) If $(i < n)$
   $i \leftarrow i + 1$
   Goto Step 2
7) Else
   exit()

The vertices may have assigned labels by n! ways
… resulting to different visit orders,
… and hence, to different color classes
… and different number of colors!

Coloring

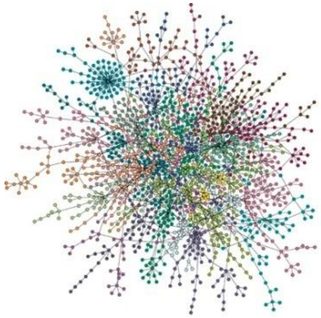


- **Approximate Coloring Algorithms**
  - **The Problem:**
    - Finding the chromatic number of a random graph is an intractable problem.
    - For small graphs exhaustive search may find this number, but for large graphs any try is ineffective.
    - Therefore, there have been proposed approximate algorithms.
      1. Serial Graph Coloring
      2. Largest First
      3. Smallest Last (Matula – Marble – Isaacson)
      4. Color Degree Method (Brelaz)

Heuristic Methods:

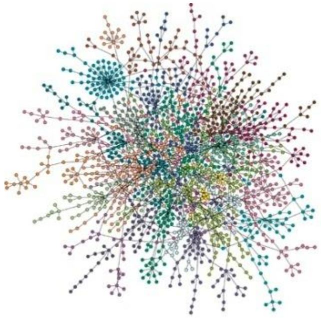
a) Vertices of hgreater degree are harder to be colored
b) Edges with the same neighboring vertices have to be colored by the same color
c) The coloring of multiple vertices with the same color is profitable.

COLORING



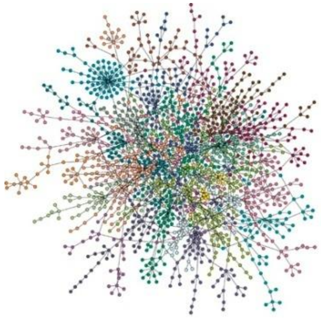
- **Approximate Coloring Algorithms**
  - **The Problem:**
    - Finding the chromatic number of a random graph is an intractable problem.
    - For small graphs exhaustive search may find this number, but for large graphs any try is ineffective.
    - Therefore, there have been proposed approximate algorithms.
      1. Serial Graph Coloring
      2. Largest First
      3. Smallest Last (Matula – Marble – Isaacson)
      4. Color Degree Method (Brelaz)

Coloring



- **Approximate Coloring Algorithms**
  - **The Problem:**
    - Finding the chromatic number of a random graph is an intractable problem.
    - For small graphs exhaustive search may find this number, but for large graphs any try is ineffective.
    - Therefore, there have been proposed approximate algorithms.
      1. Serial Graph Coloring
      2. **Largest First**
      3. Smallest Last (Matula – Marble – Isaacson)
      4. Color Degree Method (Brelaz)

Coloring

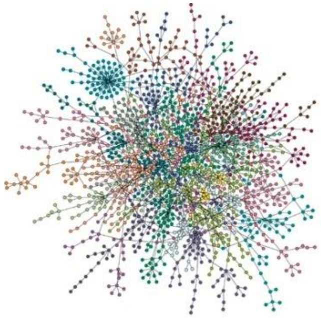


- **Approximate Coloring Algorithms**
  - **Largest First**
    - Initially the vertices are sorted in descending order w.r.t. their degree
    - Assign colors to the nodes using the serial method
      - First assign color to the vertex with the maximum degree, then to the second etc.
  - **Theorem 18 (Welsh and Powell 1967):**

    If in a connected graph G with vertex set $V(G) = \{v_1, \dots, v_n\}$, where $d(v_i) \geq d(v_{i+1})$ for $i = 1, \cdots, n-1$ we apply the serial coloring algorithm, then it holds:

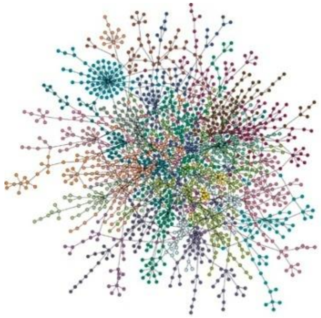
$$x(G) \leq max\big(min(i, d(v_i) + 1)\big), \forall\, i$$

COLORING



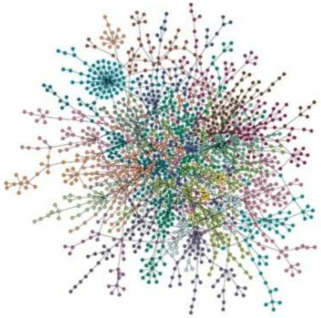
- **Approximate Coloring Algorithms**
  - **The Problem:**
    - Finding the chromatic number of a random graph is an intractable problem.
    - For small graphs exhaustive search may find this number, but for large graphs any try is ineffective.
    - Therefore, there have been proposed approximate algorithms.
      1. Serial Graph Coloring
      2. Largest First
      3. Smallest Last (Matula – Marble – Isaacson)
      4. Color Degree Method (Brelaz)

COLORING



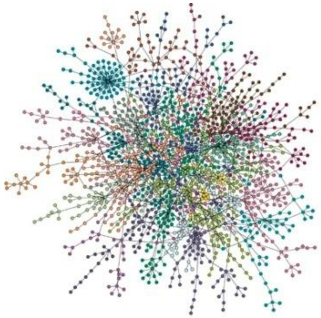
- **Approximate Coloring Algorithms**
  - **The Problem:**
    - Finding the chromatic number of a random graph is an intractable problem.
    - For small graphs exhaustive search may find this number, but for large graphs any try is ineffective.
    - Therefore, there have been proposed approximate algorithms.
      1. Serial Graph Coloring
      2. Largest First
      3. **Smallest Last (Matula – Marble – Isaacson)**
      4. Color Degree Method (Brelaz)

Coloring



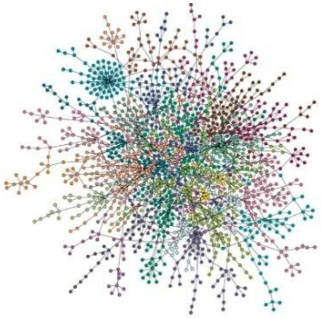
- **Approximate Coloring Algorithms**
  - **Smallest First (Matula – Marble – Isaacson 1972)**
    - Initially the vertices are sorted in descending order w.r.t. their degree
    - The vertex with the smallest degree is deleted from the graph, alongside with its adjacent vertices, as to be colored last.
    - Update the degrees of the remaining vertices and repeat the procedure.
      - The order is not necessarily the same with the one produced by the Largest First algorithm.
  - However, cases of tie (w.r.t. the degree of vertices) are treated at random.
    - Combining the two previous approaches results to a more effective solution that is called "color-degree method".
    - "Color degree" of a vertex v refers to the number of colors used for the coloring of its neighboring vertices.

Coloring



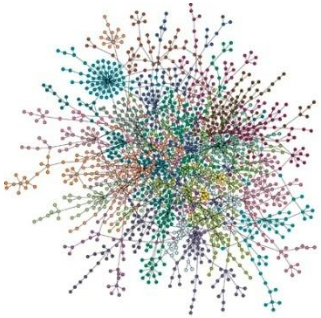
- **Approximate Coloring Algorithms**
  - **The Problem:**
    - Finding the chromatic number of a random graph is an intractable problem.
    - For small graphs exhaustive search may find this number, but for large graphs any try is ineffective.
    - Therefore, there have been proposed approximate algorithms.
      1. Serial Graph Coloring
      2. Largest First
      3. Smallest Last (Matula – Marble – Isaacson)
      4. Color Degree Method (Brelaz)

Coloring



- **Approximate Coloring Algorithms**
  - **The Problem:**
    - Finding the chromatic number of a random graph is an intractable problem.
    - For small graphs exhaustive search may find this number, but for large graphs any try is ineffective.
    - Therefore, there have been proposed approximate algorithms.
      1. Serial Graph Coloring
      2. Largest First
      3. Smallest Last (Matula – Marble – Isaacson)
      4. **Color Degree Method (Brelaz)**

Coloring



Approximate Coloring Algorithms

- **Color Degree Method (Brelaz 1979)**

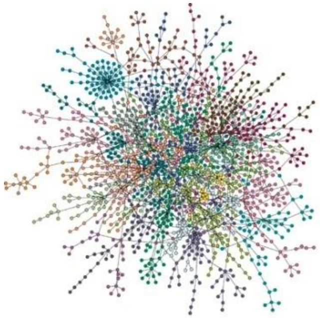
<u>Algorithm Brelaz</u>

Input: A graph $G(V,E)$

Output: A coloring of the graph.

1) Sort the vertices in descending order w.r.t. their degree
2) The vertex with the maximum degree is colored with the color "1"
3) Select the vertex with the maximum color-degree
4) If (∃ "tie")
   Select the non-colored vertex with the maximum degree in the non-colored graph.
5) Color the selected vertex with the minimum permitted color
6) If (∃ non-colored vertices)
   Goto Step 3

Coloring

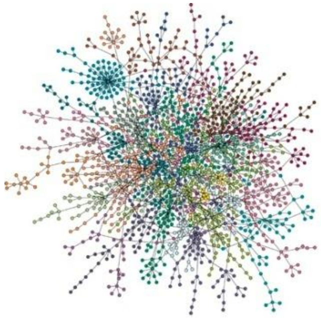


- **Approximate Coloring Algorithms**
  - **Theorem 19 (Welsh and Powell 1967):**

    If a connected graph $G, n \geq 3$ is bipartite, then the color-degree method results to the chromatic number of the graph.

    - Let a vertex v of color-degree=2
    - Let us assume that vertex v has only two adjacent vertices with different color.
    - Utilizing sequentially only these colors there can be constructed two different paths from these vertices.
    - Since the graph is connected it should contain a cycle.
    - Since the graph is bipartite, the cycle should have even length.
    - Thus, the adjacent vertices of v should have the same color, that is a contradiction.

COLORING



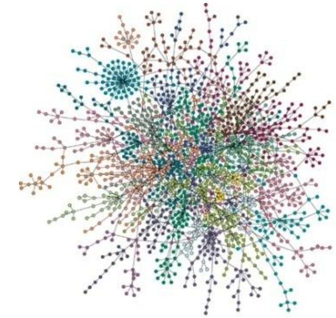
Problems and Applications

- **Course Schedule**

  We need to construct a weekly time schedule for our school with the minimum total time under the following requirements:

  1. Teacher X_i $(1 \leq \mathrm{i} \leq \mathrm{m})$ teaches course Y_j $(1 \leq \mathrm{i} \leq \mathrm{n})$ for p_{ij} hours per week.
  2. In a specific hour, a teach may teach only one course, while each course can be taught by only one teacher.
  3. No teacher teaches more that p hours per week, while no course is taught more than p hours per week.

Coloring



Problems and Applications

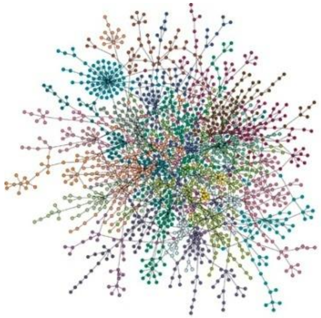
- **Course Schedule**

We need to construct a weekly time schedule for our school with the minimum total time under the following requirements:

1. Teacher X_i $(1 \leq \text{i} \leq \text{m})$ teaches course Y_j $(1 \leq \text{i} \leq \text{n})$ for p_{ij} hours per week.
2. In a specific hour, a teach may teach only one course, while each course can be taught by only one teacher.
3. No teacher teaches more that p hours per week, while no course is taught more than p hours per week.

- The corresponding graph G is $K_{n,m}$, while vertex X_i is connected with vertex Y_j through p_{ij} edges.
- The graph needs to be colored with the less colors possible.
- Through Th. 12, since the graph is bipartite, it holds that $x'(K_{m,n}) = p$. Hence, the total duration of the schedule is p hours.
- The chromatic polynomial defines the number of ways this schedule can be constructed.

Coloring



Problems and Applications

- **Coloring Bipartite Graphs**

The problem of graph coloring in the general case is difficult, however in the case of bipartite graphs, it is only needed to detect that the graph is bipartite in order to decide that only two colors are needed for its coloring.

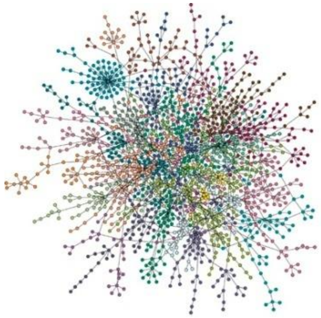
<u>Algorithm Bipartite Graph Coloring</u>

Input: A graph $G(V, E)$

Output: A 2-coloring of the graph if the graph is bipartite.

1) Select arbitrarily a vertex $v \in V$ and color it with color “1”
2) Enqueue(Q, v)
3) While(!isEmpty(Q))
    1. $u \leftarrow$ Dequeue(Q)
    2. $s \leftarrow N(v)$
    3. If($\exists\ v \in S$ with the same color as u)
        i. print(“non bipartite graph”)
        i. color each vertex in S with different color of u.
        ii. Enqueue (Q, v), $\forall\ v \in S$

COLORING



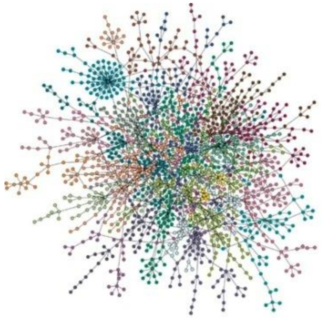
The Coloring Problem is NP-Complete

- **k-Coloring**

  Given a graph $G = (V, E)$, find out if there is proper coloring of G using $\leq\ k$ colors.

  - k-Coloring is a function $f\colon V \longrightarrow \{1, 2, \dots, k\}$ such that for each edge (x, y) of G it holds $f\ (x) \longrightarrow f\ (y)$.
  - If there exists such a function f for a given graph G, then G is k-colorable.
  - How can we decide if a graph is 2-colorable?

Coloring



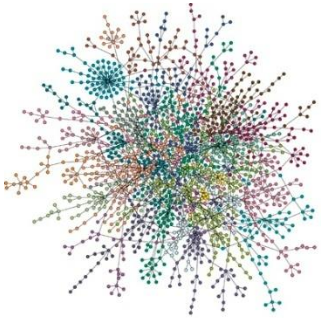
- **The Coloring Problem is NP-Complete**
  - **k-Coloring**

    Given a graph $G = (V, E)$, find out if there is proper coloring of G using $\leq$ k colors.

    - k-Coloring is a function f: $V \longrightarrow \{1, 2, \dots, k\}$ such that for each edge (x, y) of G it holds $f\ (x) \longrightarrow f\ (y)$.
    - If there exists such a function f for a given graph G, then G is k-colorable.
    - How can we decide if a graph is 2-colorable?

    … find if the graph is Bipartite

Coloring



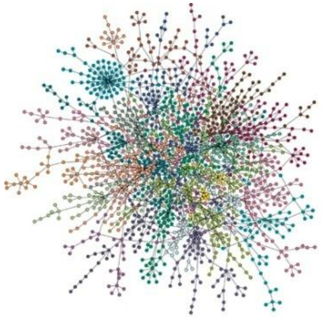
- **The Coloring Problem is NP-Complete**
  - **k-Coloring**

    Given a graph $G = (V, E)$, find out if there is proper coloring of G using $\leq\ k$ colors.

    - k-Coloring is a function $f\colon V \longrightarrow \{1, 2, \ldots, k\}$ such that for each edge (x, y) of G it holds $f\ (x)\ \longrightarrow f\ (y)$.
    - If there exists such a function f for a given graph G, then G is k-colorable.
    - How can we decide if a graph is 2-colorable?

    … however, for $k \geq\ 3$ the problem is NP-complete!!!

COLORING



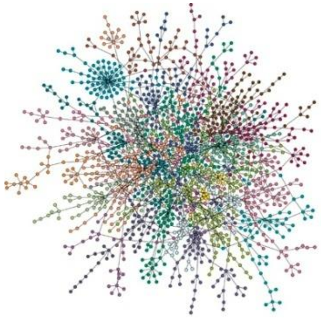
The Coloring Problem is NP-Complete

- **k-Coloring**

  Given a graph $G = (V, E)$, find out if there is proper coloring of G using $\leq\ k$ colors.

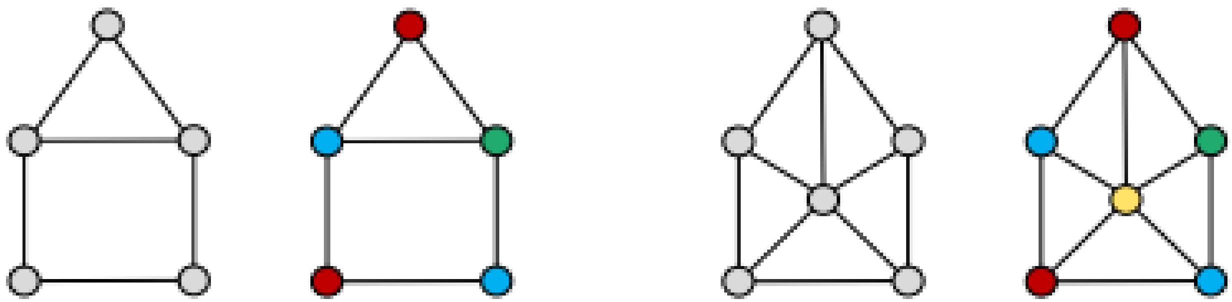
  - k-Coloring is a function $f\colon V \longrightarrow \{1, 2, \dots, k\}$ such that for each edge (x, y) of G it holds $f\ (x) \longrightarrow f\ (y)$.
  - If there exists such a function f for a given graph G, then G is k-colorable.
  - How can we decide if a graph is 2-colorable?
  - One way to handle NP-completeness is to limit the problem to subsets of input.
  - We will show that k-Coloring problem in random graphs is NP-complete even for $k = 3$!!!

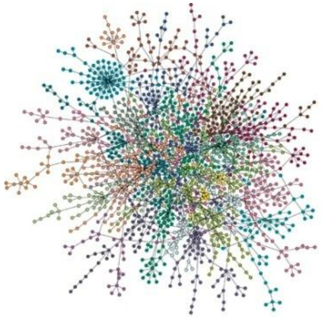
Coloring



- **The Coloring Problem is NP-Complete**
  - **3-Coloring**

    Given a graph $G = (V, E)$, find out if there is proper coloring of G using ≤ 3 colors.

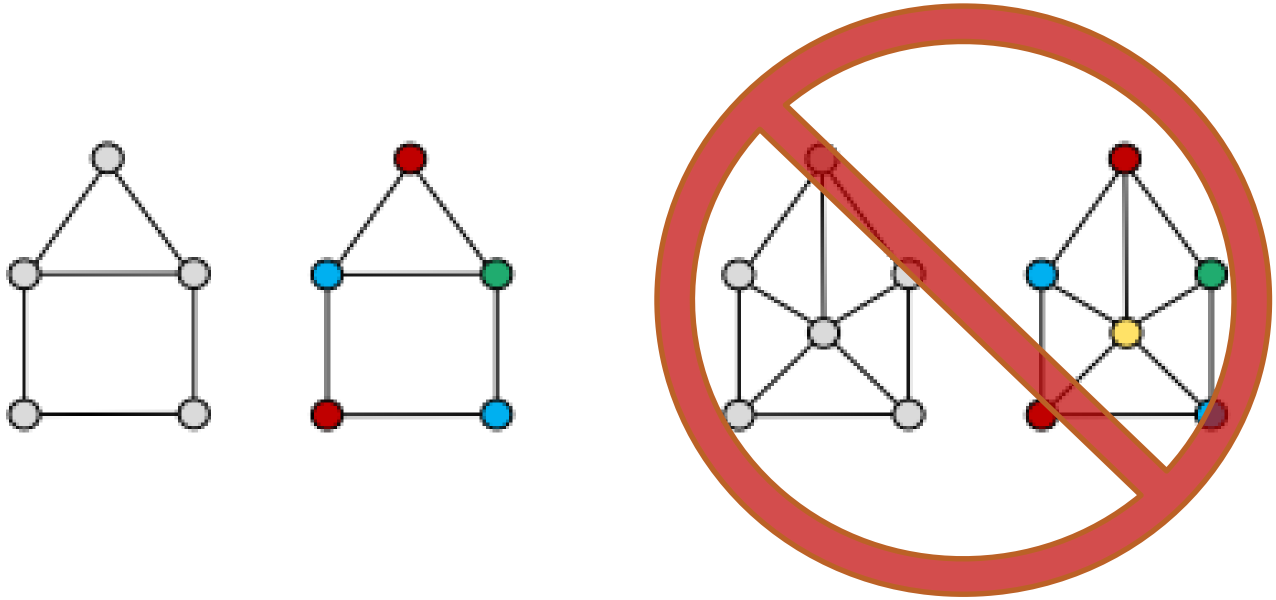




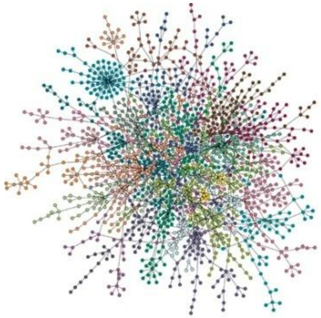
Coloring

- **The Coloring Problem is NP-Complete**
  - **3-Coloring**

    Given a graph $G = (V, E)$, find out if there is proper coloring of G using ≤ 3 colors.



Coloring



The Problem 3-Coloring and the 3SAT Problem

- **NP-complete Problem Selection**

  We need to select a problem from the "pool" of NP-Complete Problems for Reduction !!! { SAT, 3SAT, CHam, TSP, IS, Clique,… }

- **3SAT**

  Given a Boolean formula in conjunctive form with at most 3 variables in each term, find a satisfying assignment of truth values or report that there does not exist any!!!

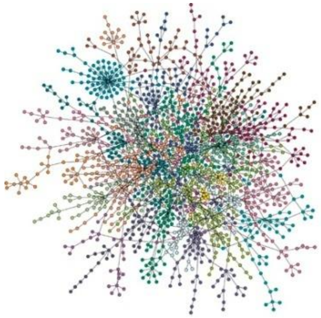
$$\boxed{(x \vee \overline{y} \vee z) \wedge (w \vee \overline{y}) \wedge (y \vee \overline{z}) \wedge (\overline{p} \vee x) \wedge (x \vee \overline{y} \vee z)}$$

- <u>We will show that</u>:

  a) 3-Coloring ∈ NP

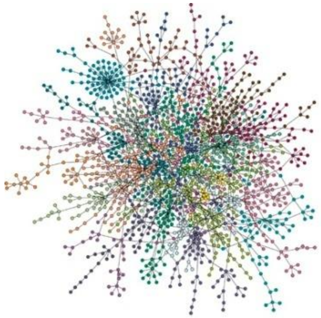
  b) 3-Coloring ∈ NP-hard: 3SAT ≤ p 3-Coloring

Coloring



- **3SAT ≤ 3-Coloring**
  - **3-Coloring ∈ NP**
    - **Certificate/Verification:** The graph G with each of its vertices colored with one color from the set {1,2,3}
    - **Verifier:** Check if for each edge (x, y) of G, the vertices x and y have different colors. (The verifier demands $O(n^2)$ time)
  - **3-Coloring ∈ NP-hard: 3SAT $\leq p$ 3-Coloring**
    - **Reduction:** Given an instance ϕ of 3SAT, we will construct in polynomial time an instance of 3-coloring (i.e., a graph G)
    - **Reduction Correctness:** The graph G is 3-colored iff the logical formula ϕ is satisfiable.

Coloring



- **3SAT ≤ 3-Coloring**
  - **3-Coloring ∈ NP**
    - **Certificate/Verification**
    - **Verifier**
  - **3-Coloring ∈ NP-hard: 3SAT $\leq p$ 3-Coloring**
    - **Reduction**
    - **Reduction Correctness**

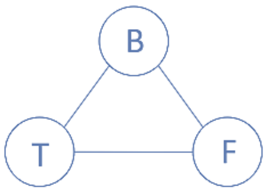
- Start from the 3SAT Problem…
- Let ϕ an instance (Boolean Formula) of 3SAT wit k terms $C_1, C_2, \dots, C_k$ (clauses) and n variables $x_1, x_2, \dots, x_n$ (variables.)
  - **Reduction:** Given an instance ϕ of 3SAT, we will construct in polynomial time an instance of 3-coloring (i.e., a graph G)
  - Construct in polynomial time an instance G of problem 3-Coloring such that:
    - If $\phi \in 3SAT$, then G ∈ 3-Coloring
    - If G ∈ 3-Colring, then $\phi \in 3SAT$

Coloring

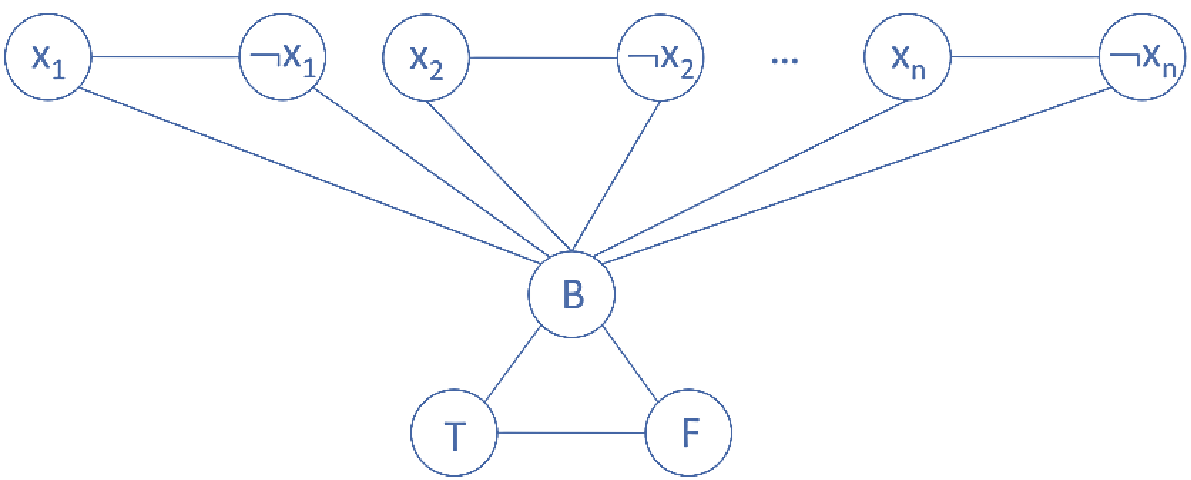
3SAT ≤ 3-Coloring – Construction of G

- For each variable x_i $(x_1, x_2, \ldots, x_n)$ of ϕ, create 2 vertices in G, one for the variable x_i and one for $\neg x_i$, and join them with an edge:

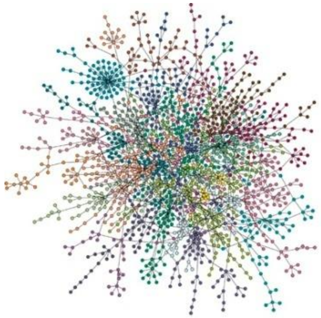
- Create 3 special nodes T, F and B and join them together:



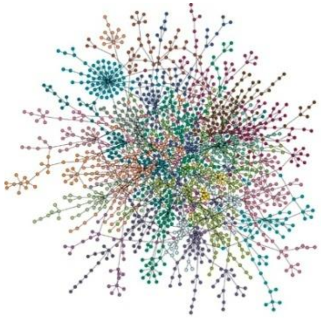
- Join each vertex-variable with vertex B:



Coloring



- **3SAT $\leq$ 3-Coloring – Construction of G**
  - Properties:
    1. Each vertex x_i and $\neg x_i$ gets different color
    2. Each vertex x_i and $\neg x_i$ takes a different color from vertex B.
    3. Vertices B, T and F take on a different color
    4. So every 3-coloring on this graph defines a valid assigning of truth values in ϕ
    5. How? Set $x_i := \text{True}$ if node x_i is colored as node T (i.e., x_i and T have the same color)

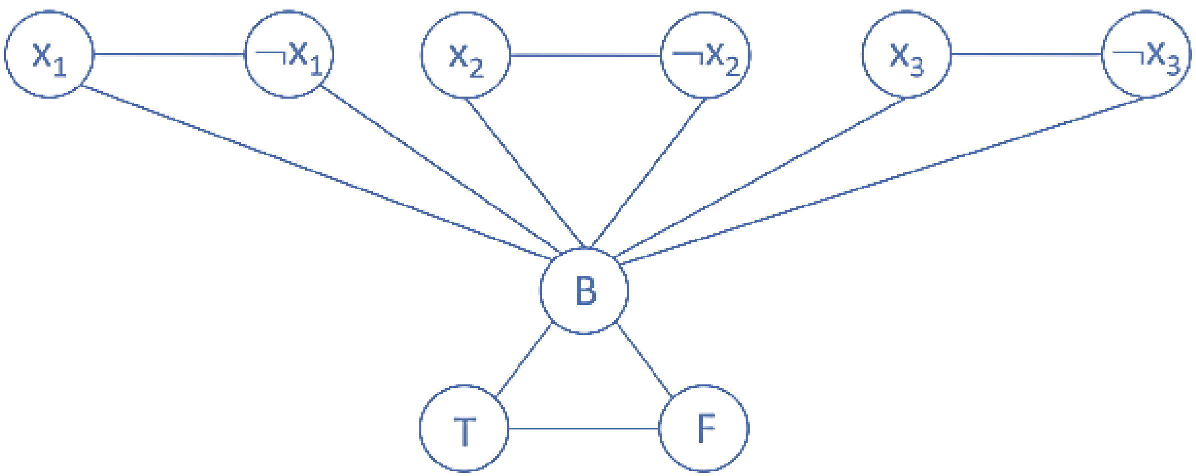


Coloring

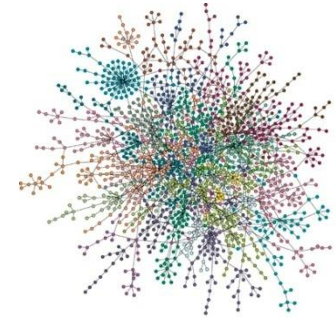
3SAT $\leq$ 3-Coloring – Construction of G

- Properties:
  1. Each vertex x_i and $\neg x_i$ gets different color
  2. Each vertex x_i and $\neg x_i$ takes a different color from vertex B.
  3. Vertices B, T and F take on a different color
  4. So every 3-color it chart specifies a valid assigning true values of ϕ
  5. How? Set $x_i :=$ True if node x_i is colored as node T (i.e., x_i and T have the same color)

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$



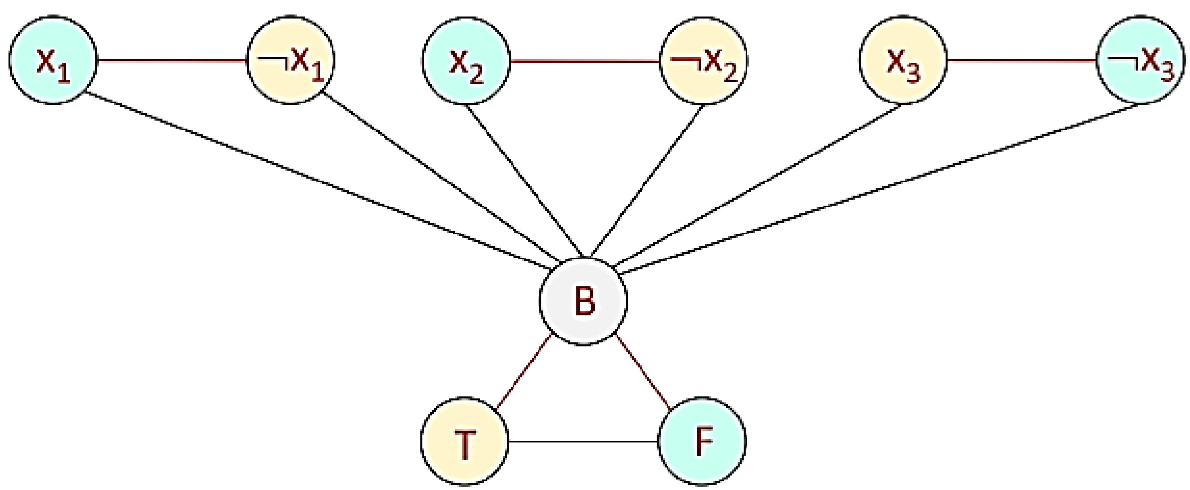
Coloring



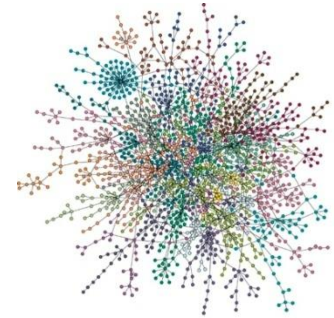
3SAT $\leq$ 3-Coloring – Construction of G

- Properties:
  1. Each vertex x_i and $\neg x_i$ gets different color
  2. Each vertex x_i and $\neg x_i$ takes a different color from vertex B.
  3. Vertices B, T and F take on a different color
  4. So every 3-color it chart specifies a valid assigning true values of ϕ
  5. How? Set $x_i :=$ True if node x_i is colored as node T (i.e., x_i and T have the same color)

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$

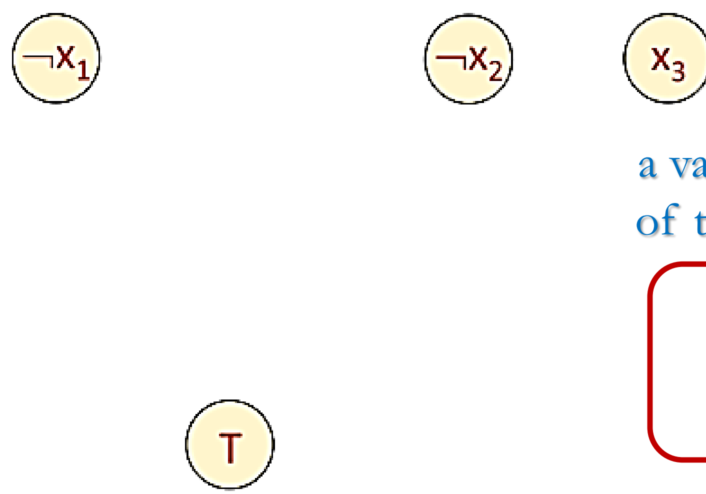


COLORING



- **3SAT ≤ 3-Coloring – Construction of G**
  - Properties:
    1. Each vertex x_i and $\neg\, x_i$ gets different color
    2. Each vertex x_i and $\neg\, x_i$ takes a different color from vertex B.
    3. Vertices B, T and F take on a different color
    4. So every 3-color it chart specifies a valid assigning true values of ϕ
    5. How? Set $x_i :=$ True if node x_i is colored as node T (i.e., x_i and T have the same color)

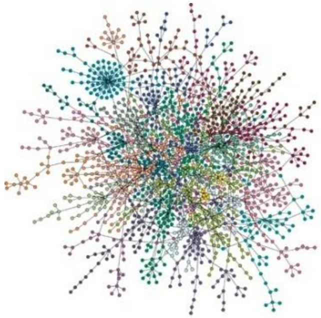
$$\varphi = (x_1 \vee x_2 \vee \neg\, x_3) \wedge (\neg\, x_2 \vee x_3 \vee \neg\, x_1)$$



a valid assignment of true values of ϕ

x_1 = False
x_2 = False
x_3 = True

Coloring



- **3SAT $\leq$ 3-Coloring – Construction of G**
  - Properties:
    1. Each vertex x_i and $\neg x_i$ gets different color
    2. Each vertex x_i and $\neg x_i$ takes a different color from vertex B.
    3. Vertices B, T and F take on a different color
    4. So every 3-color it chart specifies a valid assigning true values of ϕ
    5. How? Set $x_i :=$ True if node x_i is colored as node T (i.e., x_i and T have the same color)

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

$\neg x_1$ $\neg x_2$ x_3

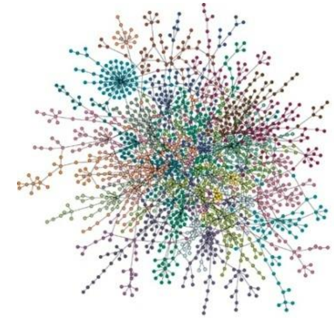
a valid assignment of true values of ϕ

Is it satisfying ???

x_1 = False
x_2 = False
x_3 = True

T

Coloring



3SAT ≤ 3-Coloring – Construction of G

- Properties:
  1. Each vertex x_i and $\neg x_i$ gets different color
  2. Each vertex x_i and $\neg x_i$ takes a different color from vertex B.
  3. Vertices B, T and F take on a different color
  4. So every 3-color it chart specifies a valid assigning true values of ϕ
  5. How? Set $x_i := $ True if node x_i is colored as node T (i.e., x_i and T have the same color)

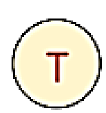
$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$

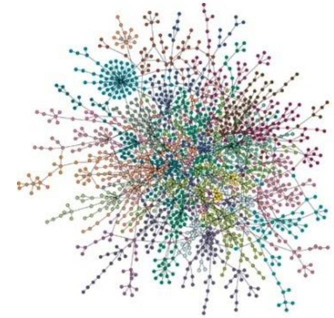
FALSE!!!

a valid assignment of true values of ϕ

Is it satisfying ???

x_1 = False
x_2 = False
x_3 = True





Coloring

3SAT ≤ 3-Coloring – Construction of G

- Properties:
  1. Each vertex x_i and $\neg x_i$ gets different color
  2. Each vertex x_i and $\neg x_i$ takes a different color from vertex B.
  3. Vertices B, T and F take on a different color
  4. So every 3-color it chart specifies a valid assigning true values of ϕ
  5. How? Set $x_i := $ True if node x_i is colored as node T (i.e., x_i and T have the same color)

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$

FALSE!!!

$\neg x_1$ $\neg x_2$ x_3

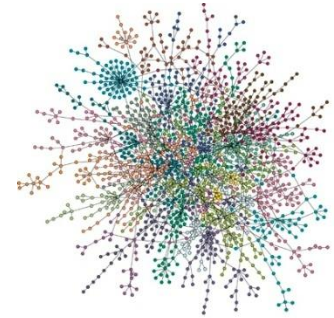
a satisfying truth value assignment of logical type ϕ

Is it satisfying ???

x_1 = True
x_2 = False
x_3 = True

T

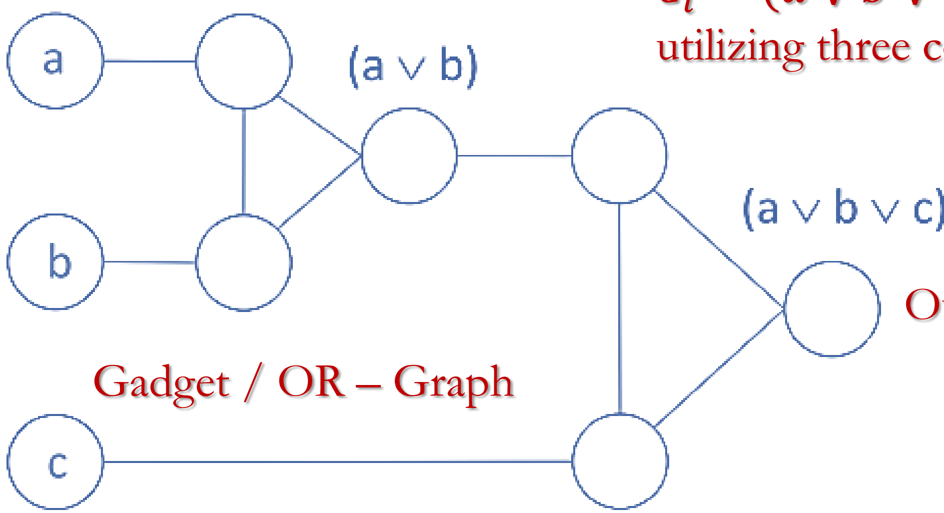
Coloring



3SAT ≤ 3-Coloring – Construction of G

- The assignment of values to the variables $x_1, x_2, \dots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \dots, C_k$ of ϕ.
  - For each term $C_i = (a \vee b \vee c)$ of ϕ, construct a gadget graph joining vertices x_p, x_q and x_r of G' that correspond to the variables $a, b,$ and c and implements the logical OR.

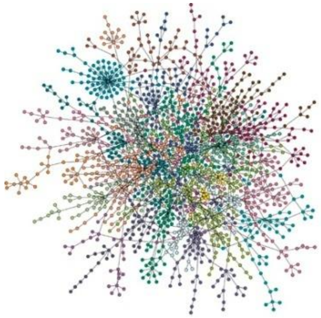
$C_i = (a \vee b \vee c)$ … the OR is modeled utilizing three colores, namely {T,F,B}



Output Vertex

Gadget / OR – Graph

Coloring

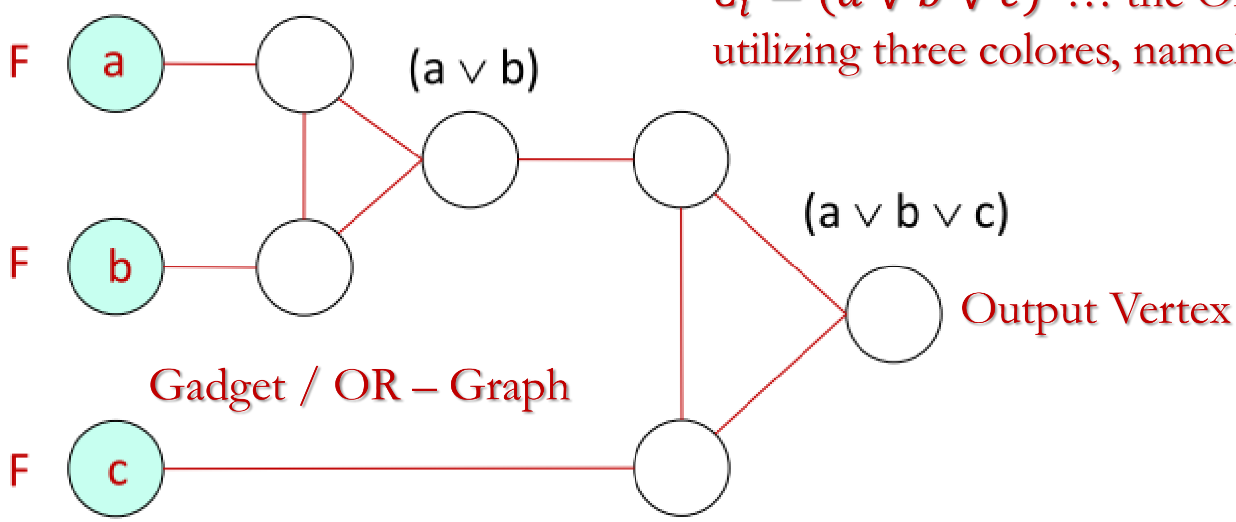


3SAT ≤ 3-Coloring – Construction of G

- The assignment of values to the variables $x_1, x_2, \ldots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \ldots, C_k$ of ϕ.

  - For each term $C_i = (a \vee b \vee c)$ of ϕ, construct a gadget graph joining vertices x_p, x_q and x_r of G' that correspond to the variables $a, b,$ and c and implements the logical OR.

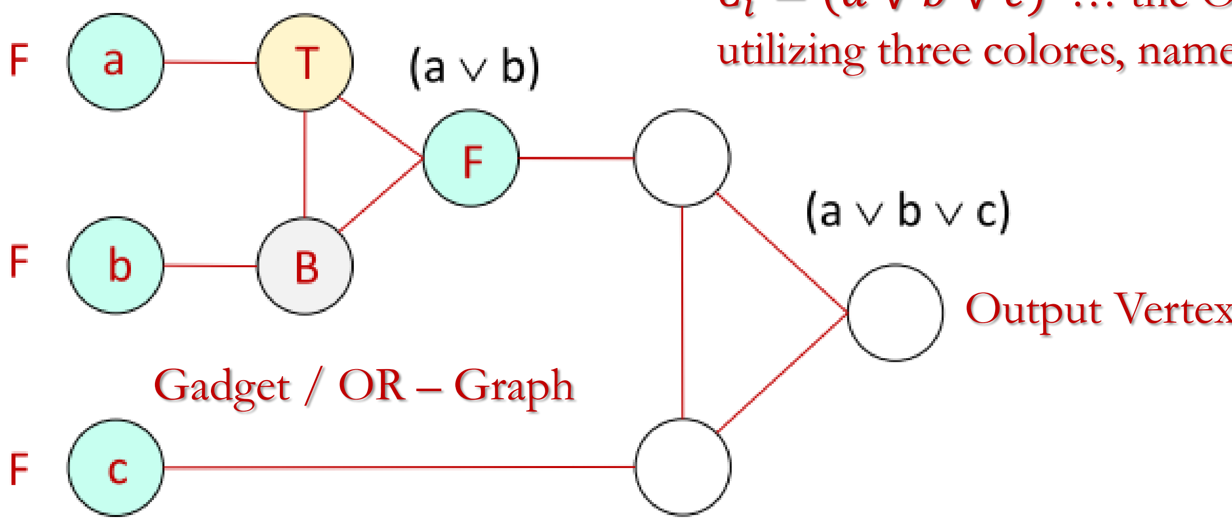
$C_i = (a \vee b \vee c)$ … the OR is modeled utilizing three colores, namely {T,F,B}



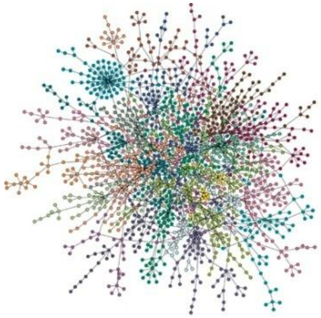
COLORING

- **$3SAT \leq$ 3-Coloring – Construction of G**
  - The assignment of values to the variables $x_1, x_2, \dots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \dots, C_k$ of ϕ.
    - For each term $C_i = (a \vee b \vee c)$ of ϕ, construct a gadget graph joining vertices x_p, x_q and x_r of G' that correspond to the variables $a, b,$ and c and implements the logical OR.

$C_i = (a \vee b \vee c)$ … the OR is modeled utilizing three colores, namely {T,F,B}



COLORING

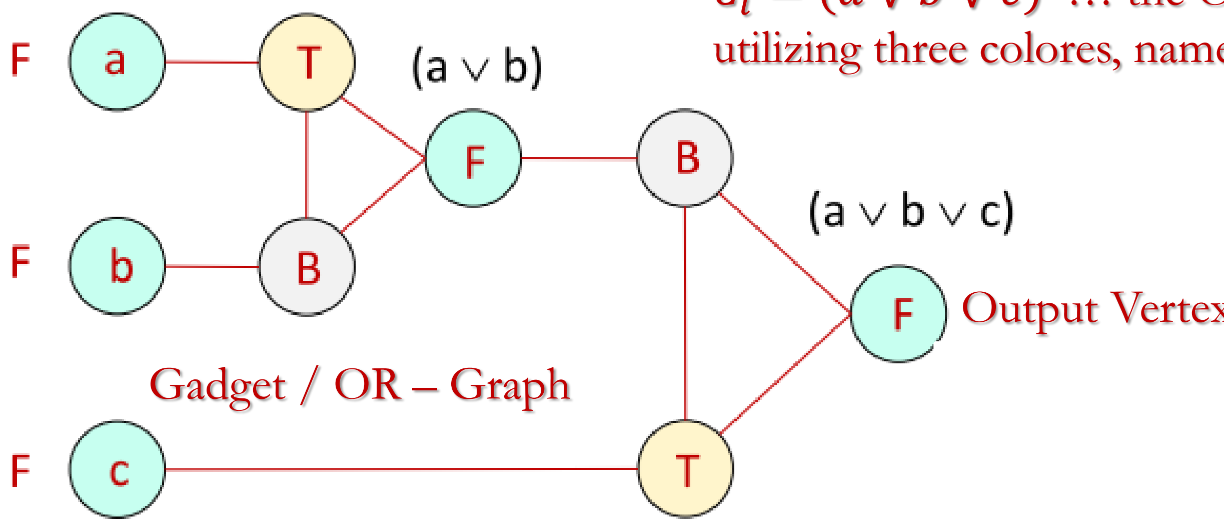


3SAT ≤ 3-Coloring – Construction of G

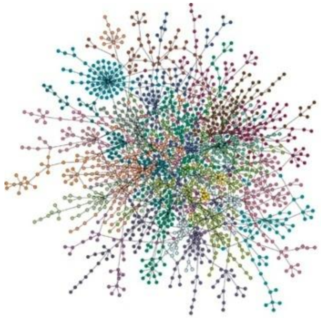
- The assignment of values to the variables $x_1, x_2, \dots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \dots, C_k$ of ϕ.

  - For each term $C_i = (a \vee b \vee c)$ of ϕ, construct a gadget graph joining vertices x_p, x_q and x_r of G' that correspond to the variables $a, b,$ and c and implements the logical OR.

$C_i = (a \vee b \vee c)$ … the OR is modeled utilizing three colores, namely {T,F,B}

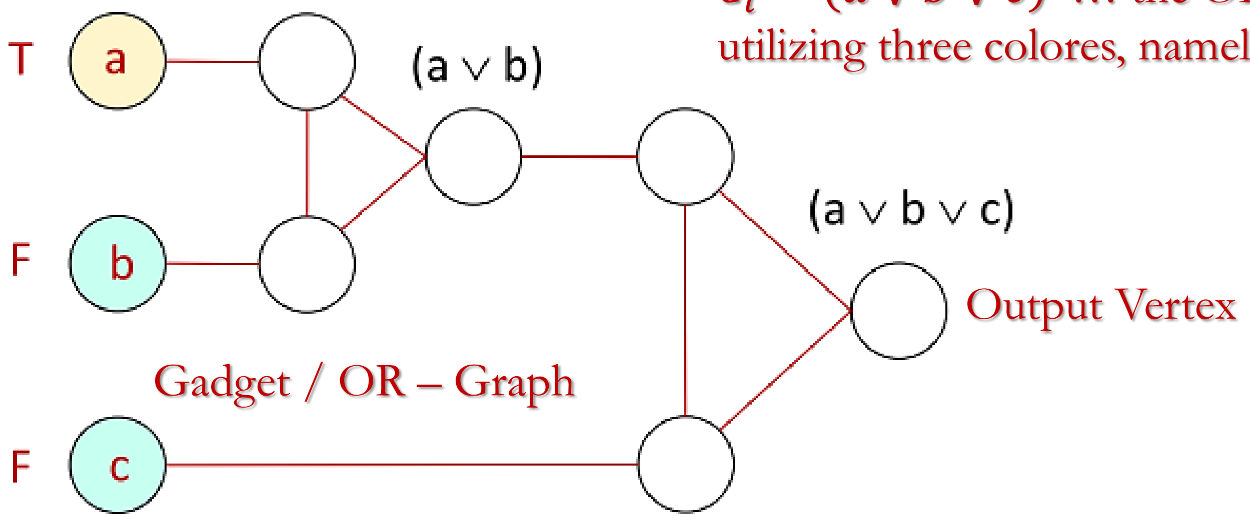


COLORING

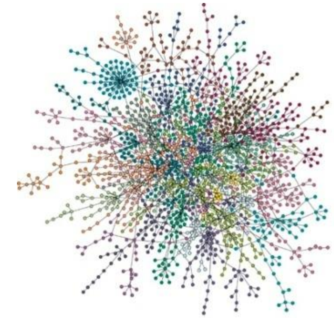


- **3SAT ≤ 3-Coloring – Construction of G**
  - The assignment of values to the variables $x_1, x_2, \ldots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \ldots, C_k$ of ϕ.
    - For each term $C_i = (a \vee b \vee c)$ of ϕ, construct a gadget graph joining vertices x_p, x_q and x_r of G' that correspond to the variables $a, b,$ and c and implements the logical OR.

$C_i = (a \vee b \vee c)$ … the OR is modeled utilizing three colores, namely {T,F,B}



COLORING

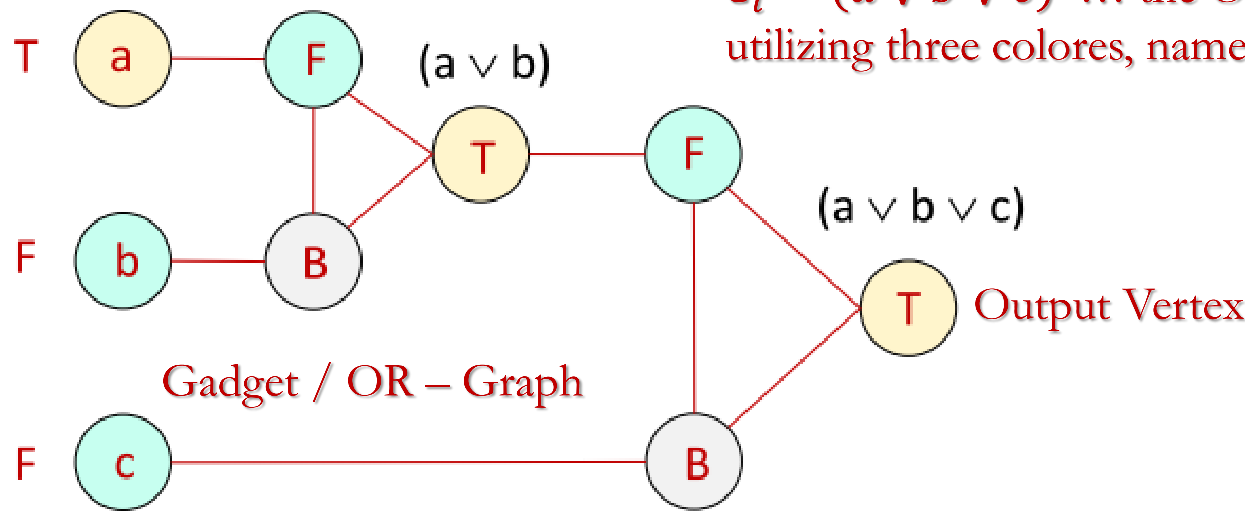


3SAT ≤ 3-Coloring – Construction of G

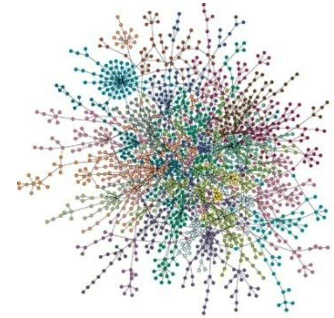
- The assignment of values to the variables $x_1, x_2, \dots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \dots, C_k$ of ϕ.

  - For each term $C_i = (a \lor b \lor c)$ of ϕ, construct a gadget graph joining vertices x_p, x_q and x_r of G' that correspond to the variables $a, b,$ and c and implements the logical OR.

$C_i = (a \lor b \lor c)$ … the OR is modeled utilizing three colores, namely {T,F,B}



Coloring

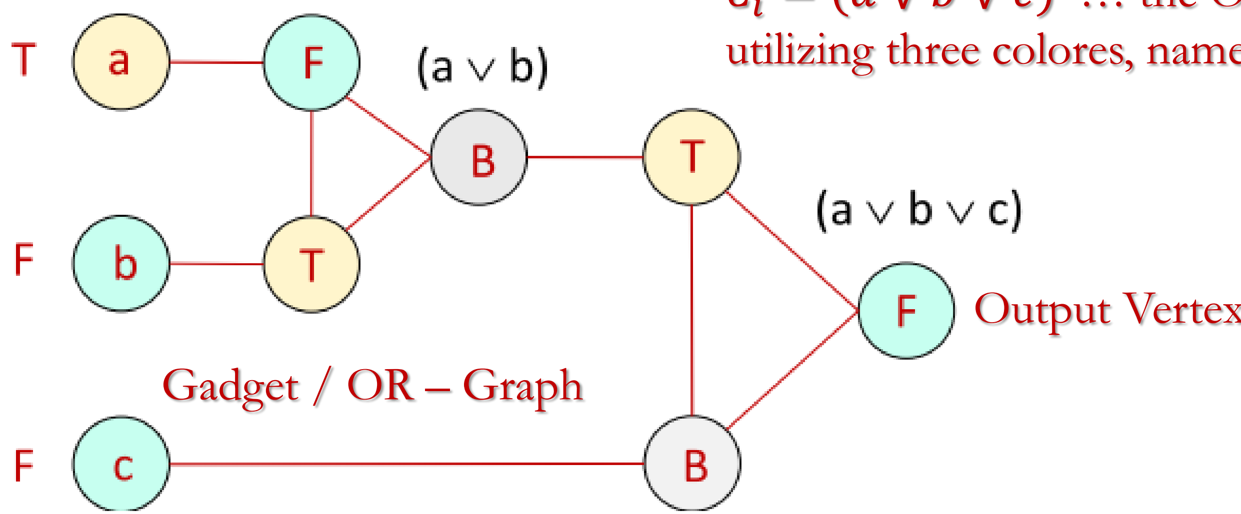


3SAT ≤ 3-Coloring – Construction of G

- The assignment of values to the variables $x_1, x_2, \dots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \dots, C_k$ of ϕ.

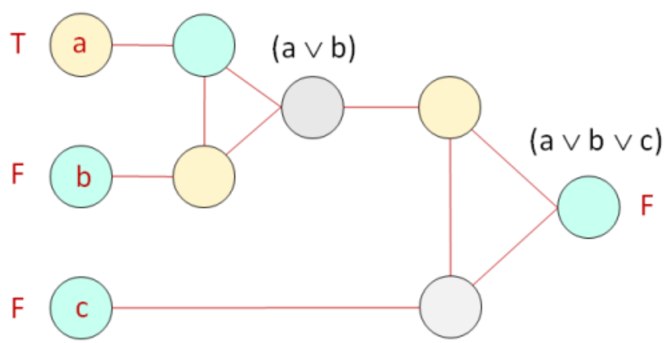
  - For each term $C_i = (a \vee b \vee c)$ of ϕ, construct a gadget graph joining vertices x_p, x_q and x_r of G' that correspond to the variables $a, b,$ and c and implements the logical OR.

$C_i = (a \vee b \vee c)$ … the OR is modeled utilizing three colores, namely {T,F,B}



Coloring

- **3SAT ≤ 3-Coloring – Construction of G**
  - The assignment of values to the variables $x_1, x_2, \ldots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \ldots, C_k$ of ϕ.
    - If all the variables a, b, c are colored with color F, then the output vertex of OR-Graph is colored with F too.



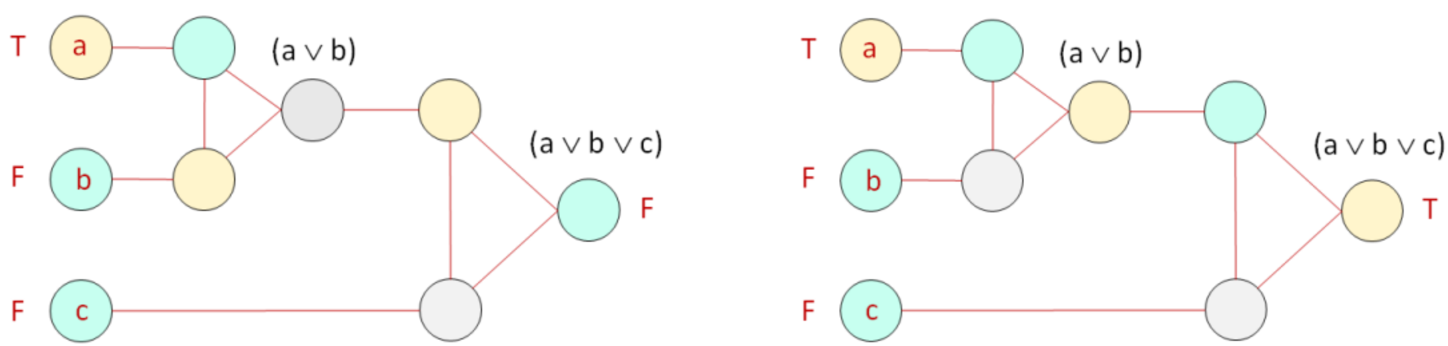
Coloring

3SAT $\leq$ 3-Coloring – Construction of G

- The assignment of values to the variables $x_1, x_2, \dots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \dots, C_k$ of ϕ.

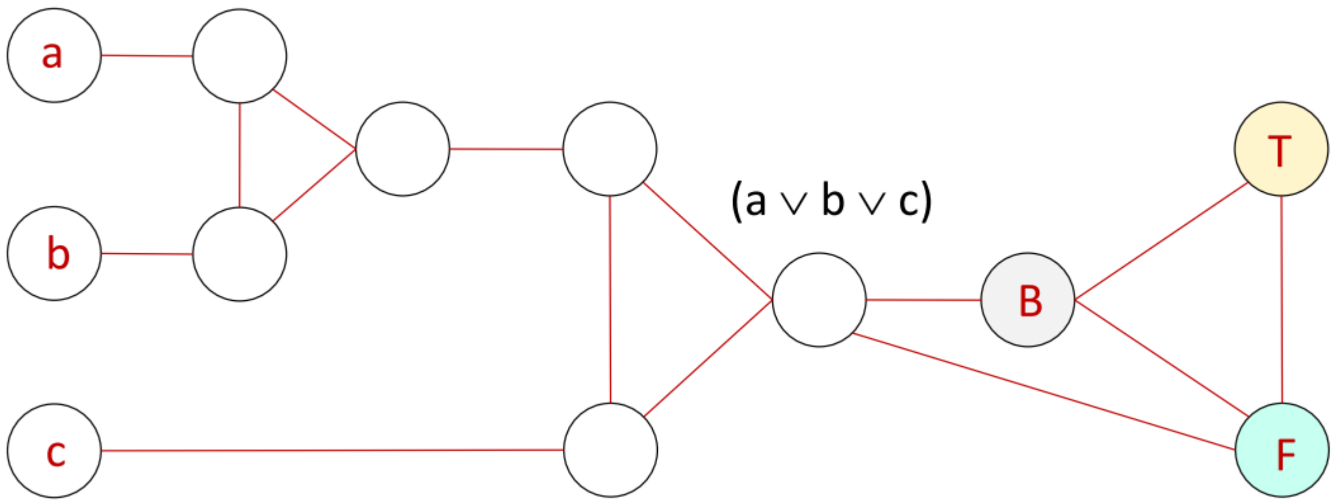
  - If all the variables a, b, c are colored with color F, then the output vertex of OR-Graph is colored with F too.

  - If at least one variable from a, b, c has color T, the there exists a proper 3-coloring of the OR-Graph, where output vertex has color T,



Coloring

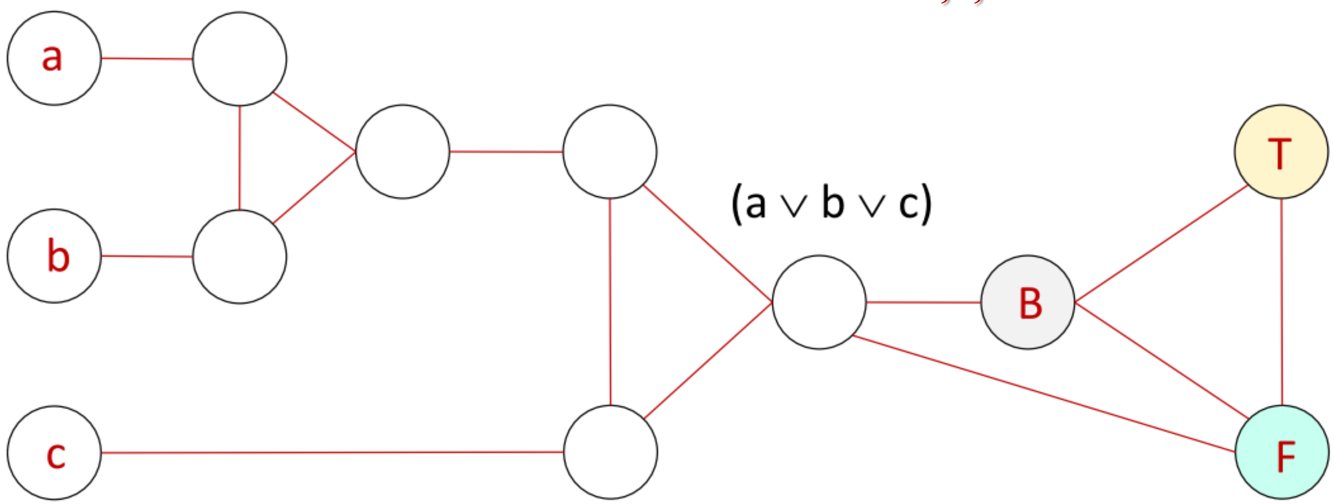
- **3SAT ≤ 3-Coloring – Construction of G**
  - The assignment of values to the variables $x_1, x_2, \dots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \dots, C_k$ of ϕ.
    - The Gadget or OR-Graph



COLORING

- **3SAT ≤ 3-Coloring – Construction of G**
  - The assignment of values to the variables $x_1, x_2, \dots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \dots, C_k$ of ϕ.
    - The Gadget or OR-Graph

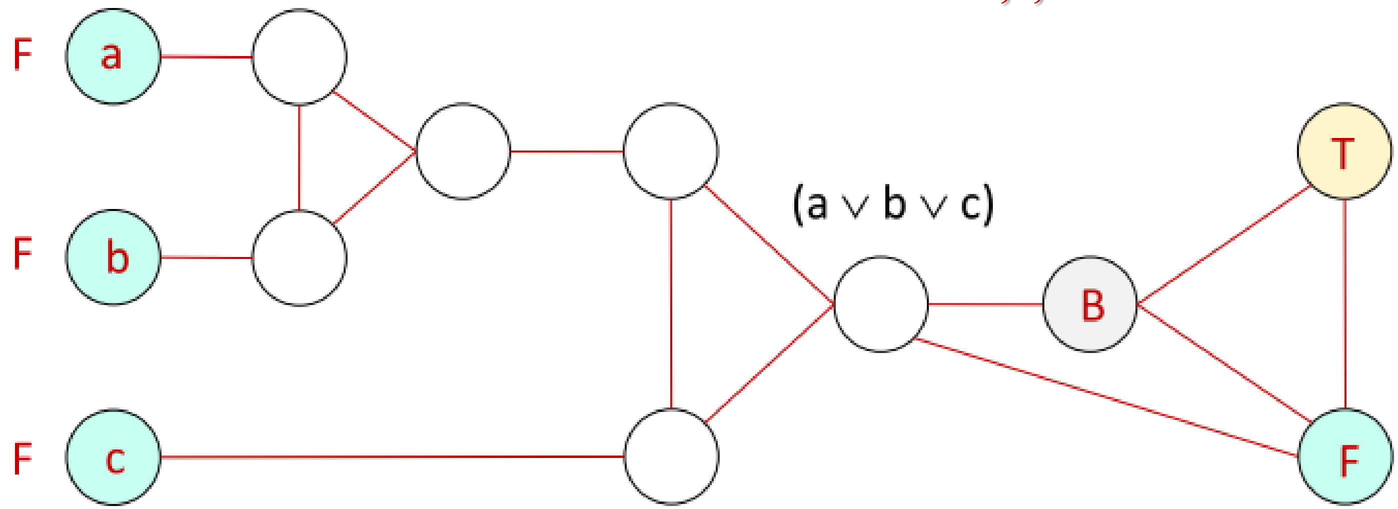
What would happen if every variable a,b,c has value FALSE?

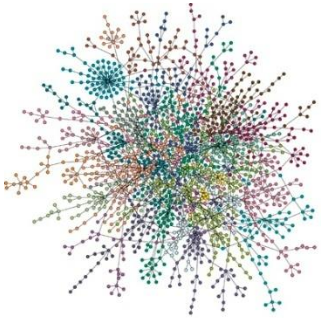


Coloring

- **3SAT ≤ 3-Coloring – Construction of G**
  - The assignment of values to the variables $x_1, x_2, \ldots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \ldots, C_k$ of ϕ.
    - The Gadget or OR-Graph

What would happen if every variable a,b,c has value FALSE?

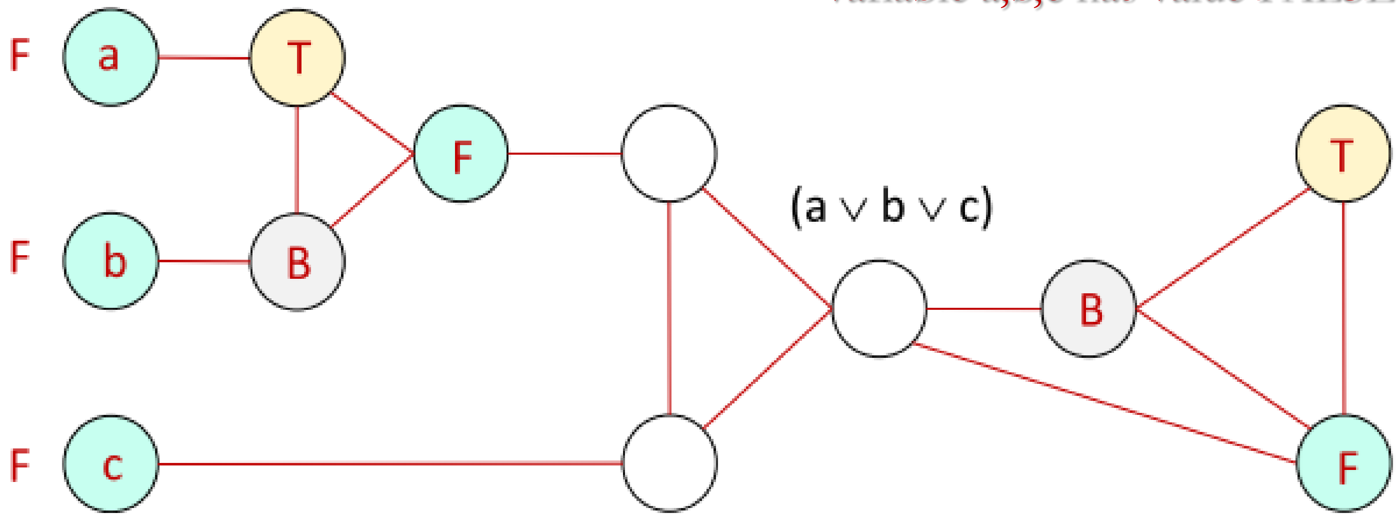




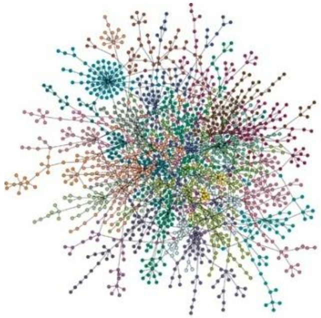
COLORING

- **3SAT ≤ 3-Coloring – Construction of G**
  - The assignment of values to the variables $x_1, x_2, \ldots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \ldots, C_k$ of ϕ.
    - The Gadget or OR-Graph

What would happen if every variable a,b,c has value FALSE?

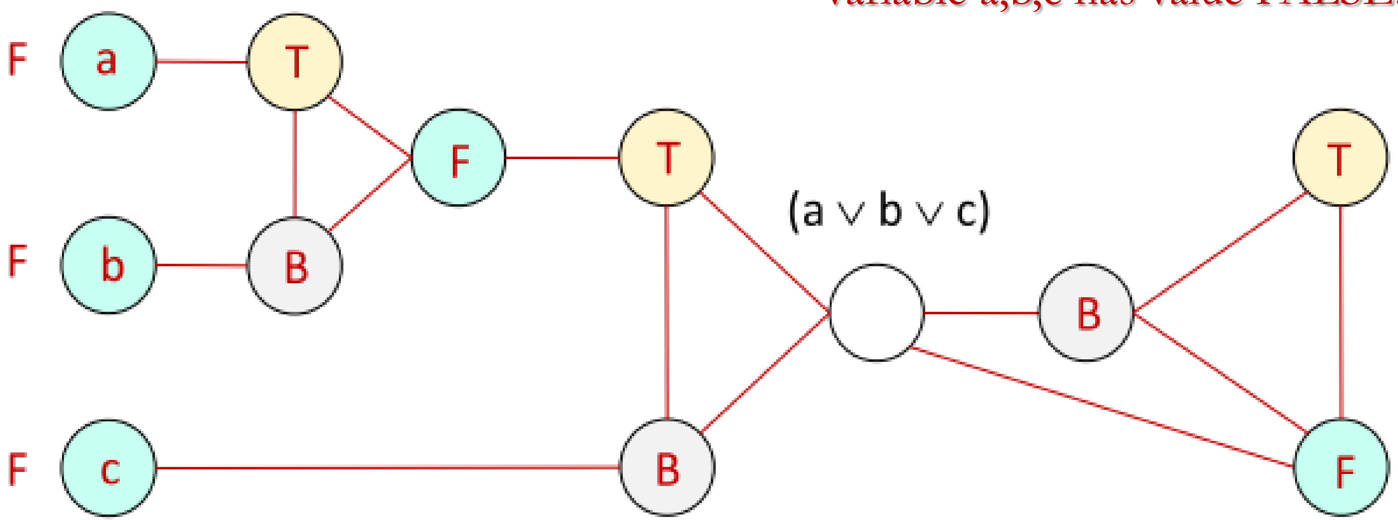


Coloring



- **3SAT ≤ 3-Coloring – Construction of G**
  - The assignment of values to the variables $x_1, x_2, \ldots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \ldots, C_k$ of ϕ.
    - The Gadget or OR-Graph

What would happen if every variable a,b,c has value FALSE?



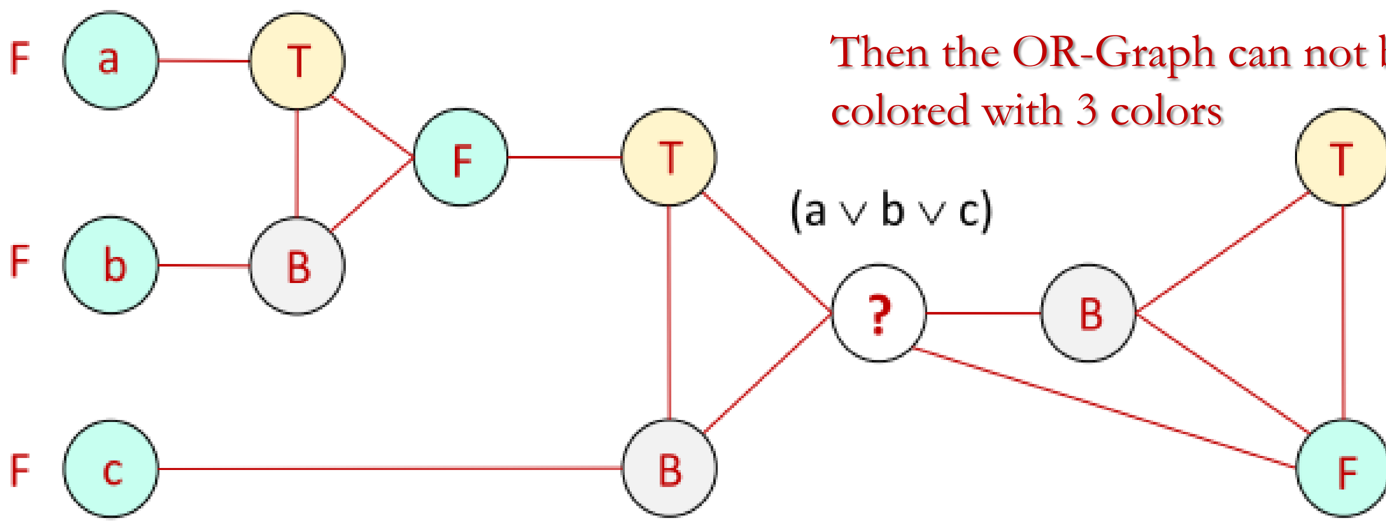
Coloring

3SAT ≤ 3-Coloring – Construction of G

- The assignment of values to the variables $x_1, x_2, \ldots, x_n$ should ensure the satisfaction of each term $C_1, C_2, \ldots, C_k$ of ϕ.
  - The Gadget or OR-Graph

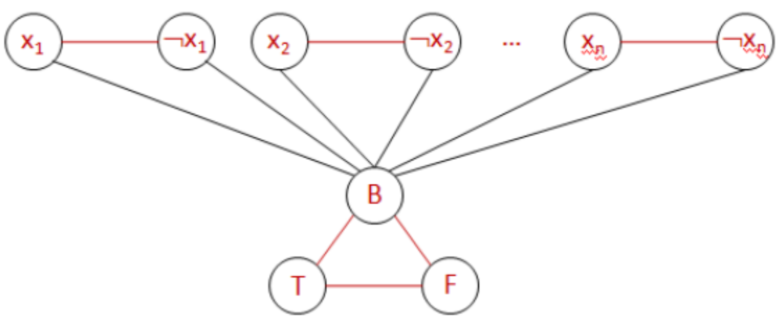
What would happen if every variable a,b,c has value FALSE?

Then the OR-Graph can not be colored with 3 colors

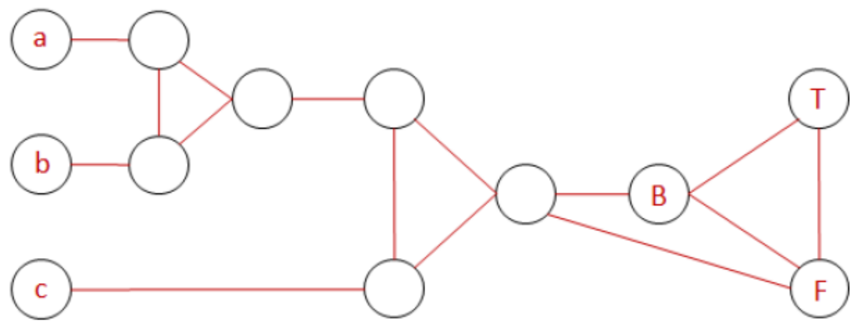


COLORING

- **3SAT ≤ 3-Coloring – Construction of G**
  - For each term $C_i = (x_p \vee x_q \vee x_r) = (a \vee b \vee c)$ of ϕ, join with edges the vertices x_p, x_q, x_r of the graph G' with the vertices $a, b, c,$ respectively, of the OR-Graph

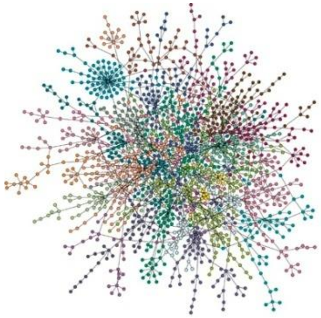


G'



$OR - Graph$

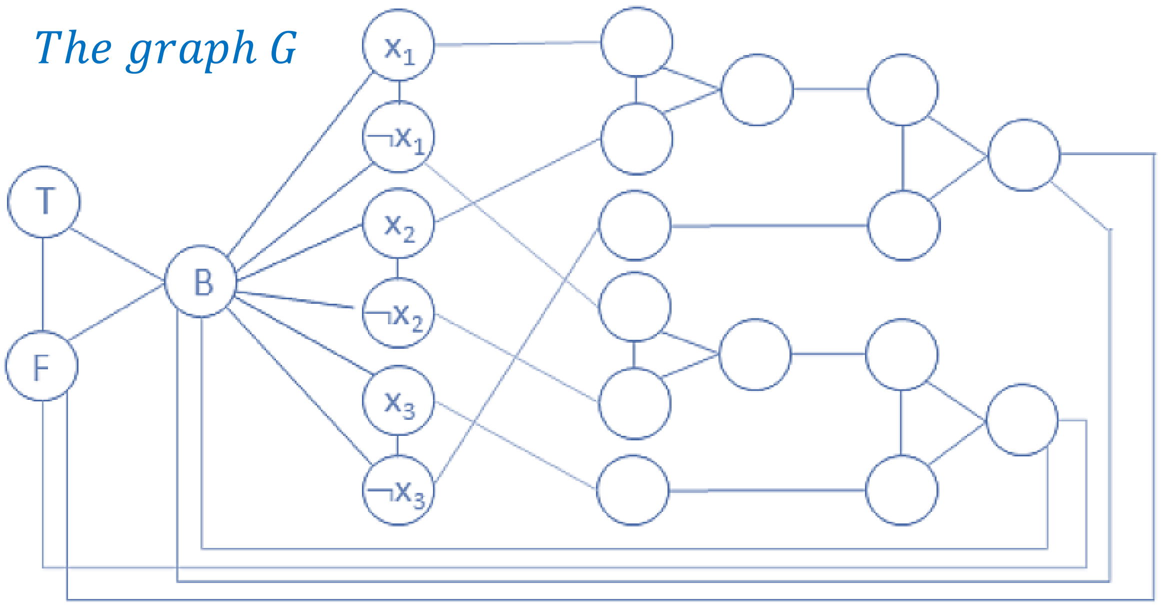
Coloring



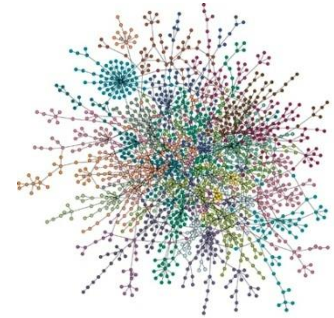
- **3SAT $\leq$ 3-Coloring – Construction of G**
  - For each term $C_i = (x_p \vee x_q \vee x_r) = (a \vee b \vee c)$ of ϕ, join with edges the vertices x_p, x_q, x_r of the graph G' with the vertices a, b, c, respectively, of the OR-Graph

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$

The graph G

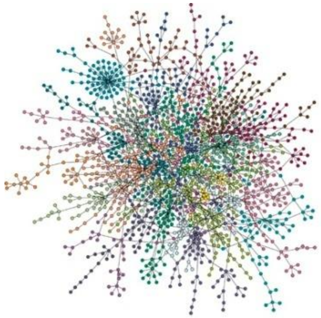


Coloring



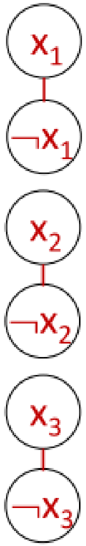
- **3SAT $\leq$ 3-Coloring – Correctness of Reduction**
  - **If ϕ Satisfying $\Rightarrow$ G is 3-Colorable**
    - If x_i = True, then color the node x_i of G with T color and the node $\neg x_i$ in color F. If x_i = False, then do the opposite.
    - In each term $C_i = (a \vee b \vee c)$ of ϕ, at least one a variable of a, b and c has True value.
    - Therefore the OR-graph corresponding to C_i can be colored with 3 colors.
    - Therefore, the graph G is colored in 3 colors.
  - **If G is 3-Colorable $\Rightarrow$ ϕ is Satisfying**
    - If node x_i has the same color as node T, then $x_i :=$ True, otherwise $x_i :=$ False (this is a valid assignment of values).
    - Let $C_i = (a \vee b \vee c)$ a term of ϕ. At least one a variable of a, b and c has a True value, because if all had a False value then the G could not be colored with 3 colors (as we have shown before).
    - So, ϕ is Satisfying.

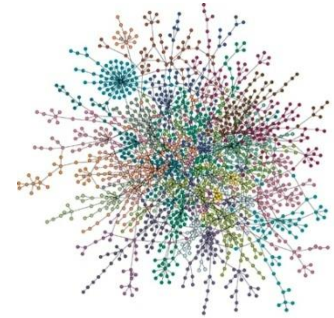
COLORING



- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT $\leq$ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$



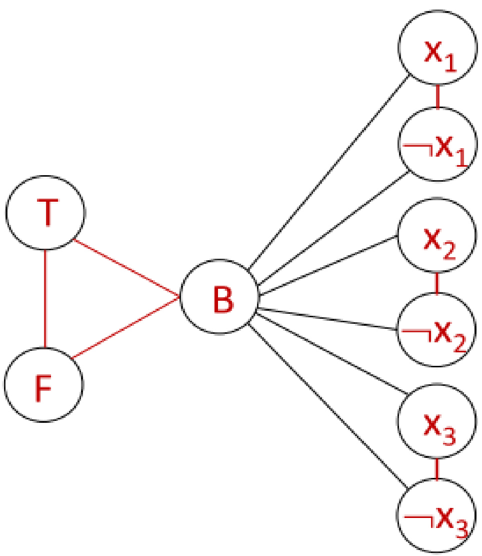


Coloring

3-Coloring Problem is NP-Complete

- **We can use the 3-Coloring problem to solve the 3SAT problem**
- **We have shown that 3SAT $\leq$ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

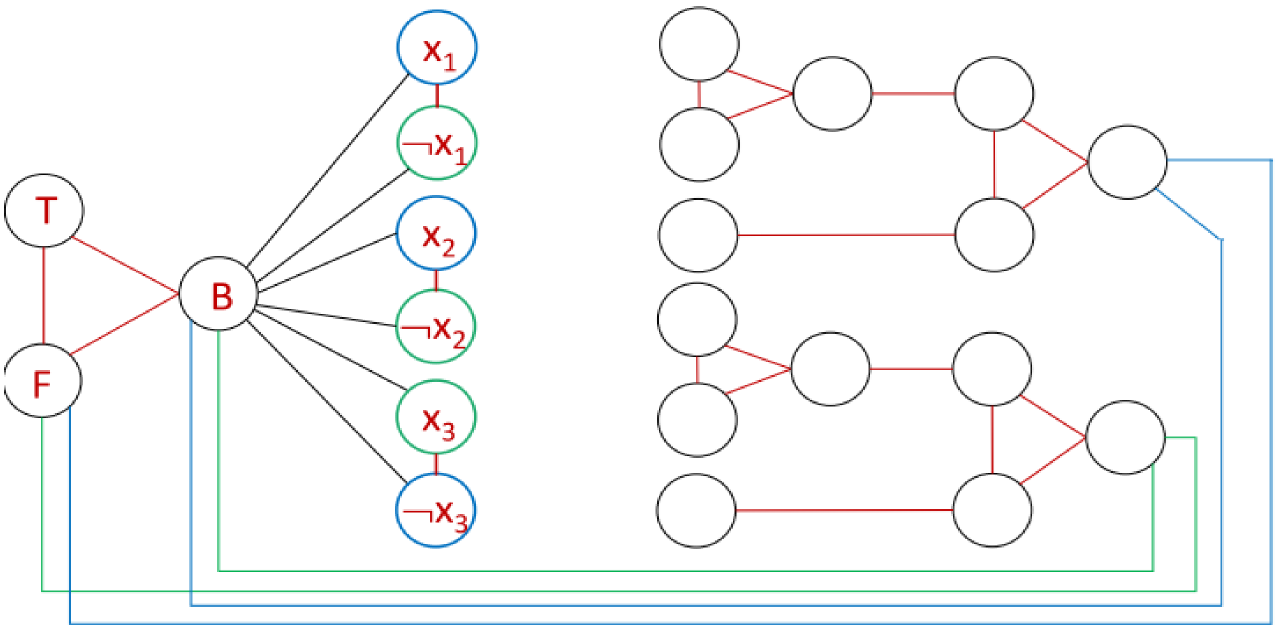
$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$



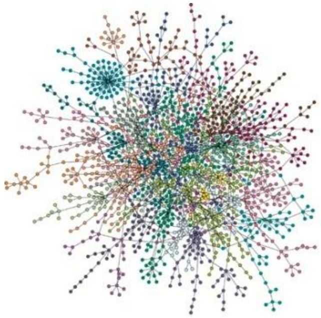
COLORING

- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT ≤ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

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

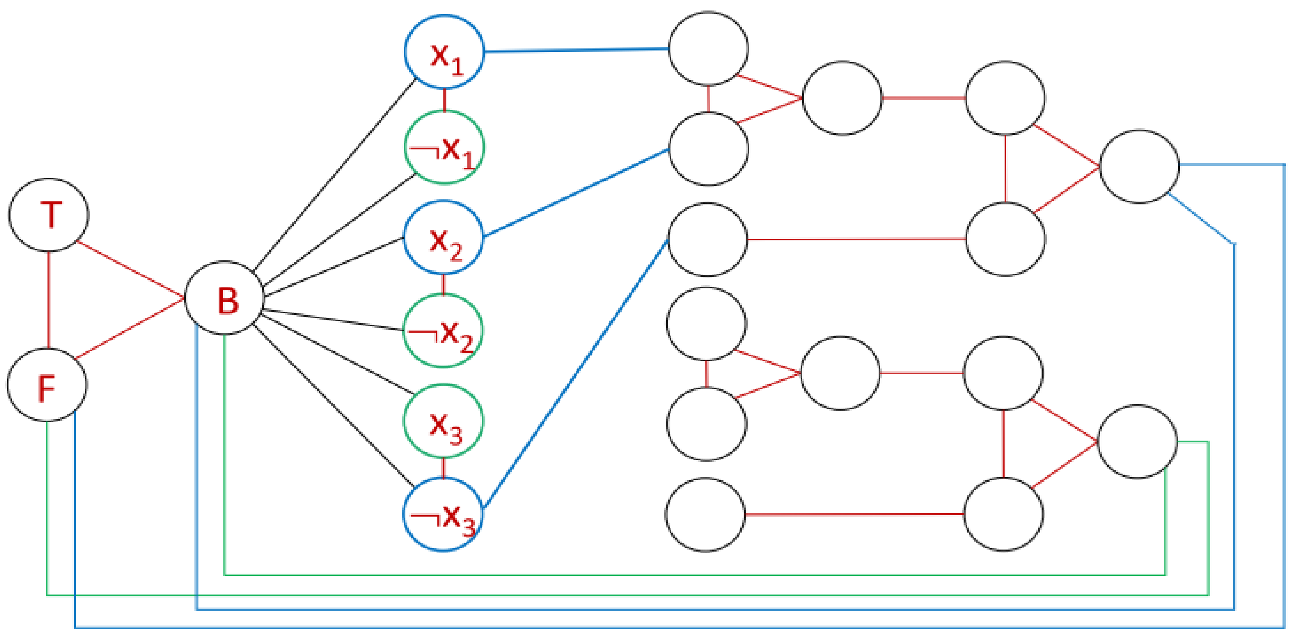


COLORING



- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT ≤ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

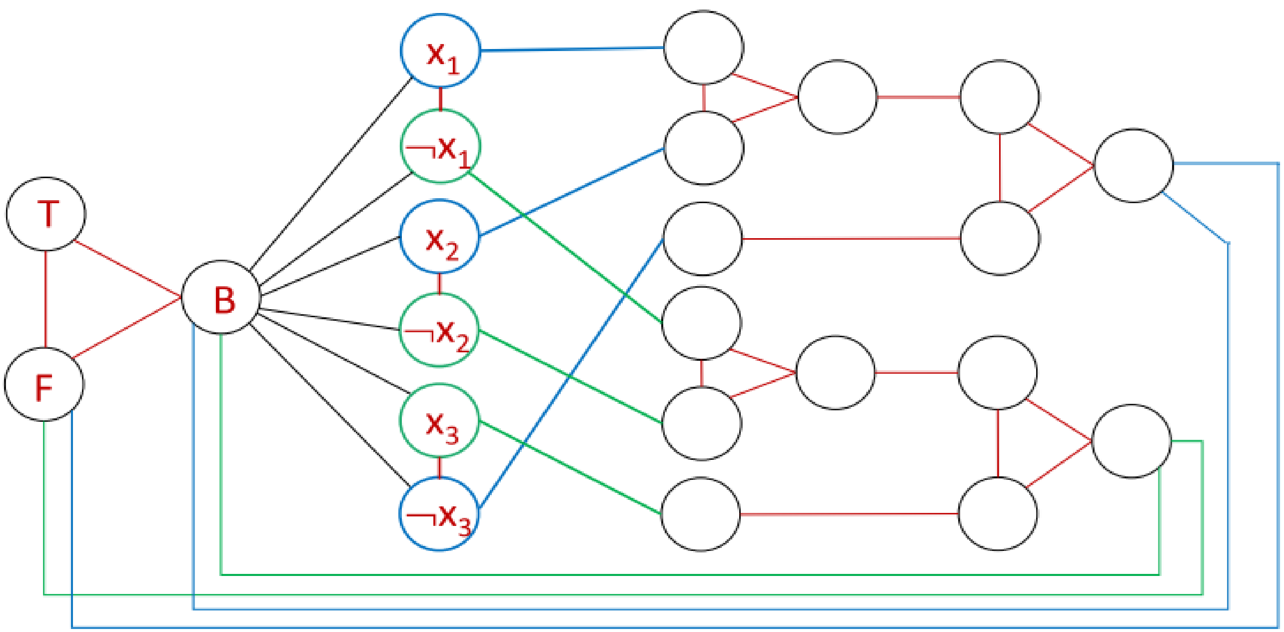
$$\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)$$

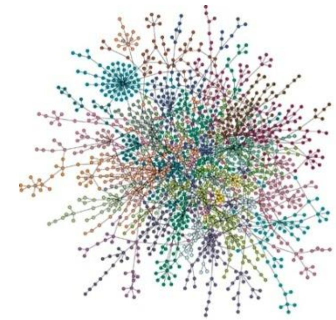


Coloring

- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT $\leq$ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$

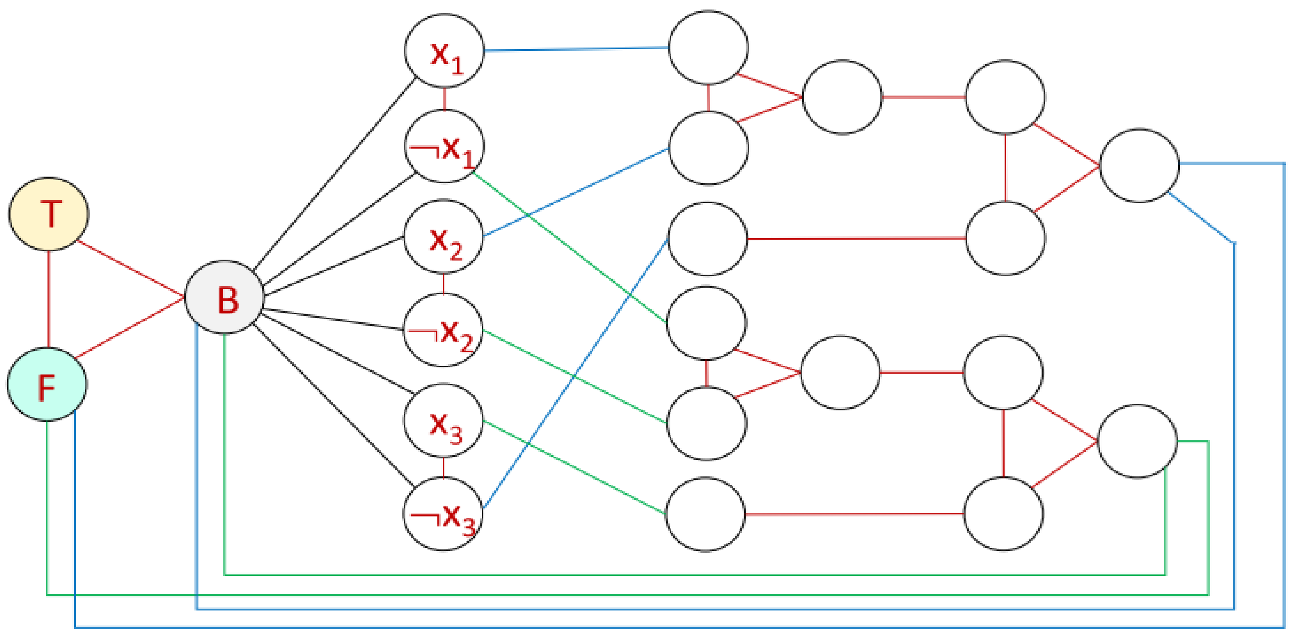




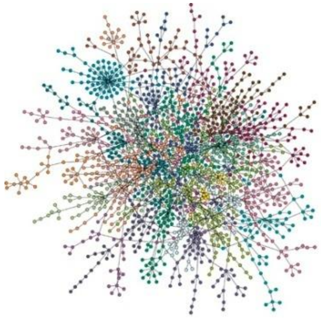
COLORING

- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT ≤ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$

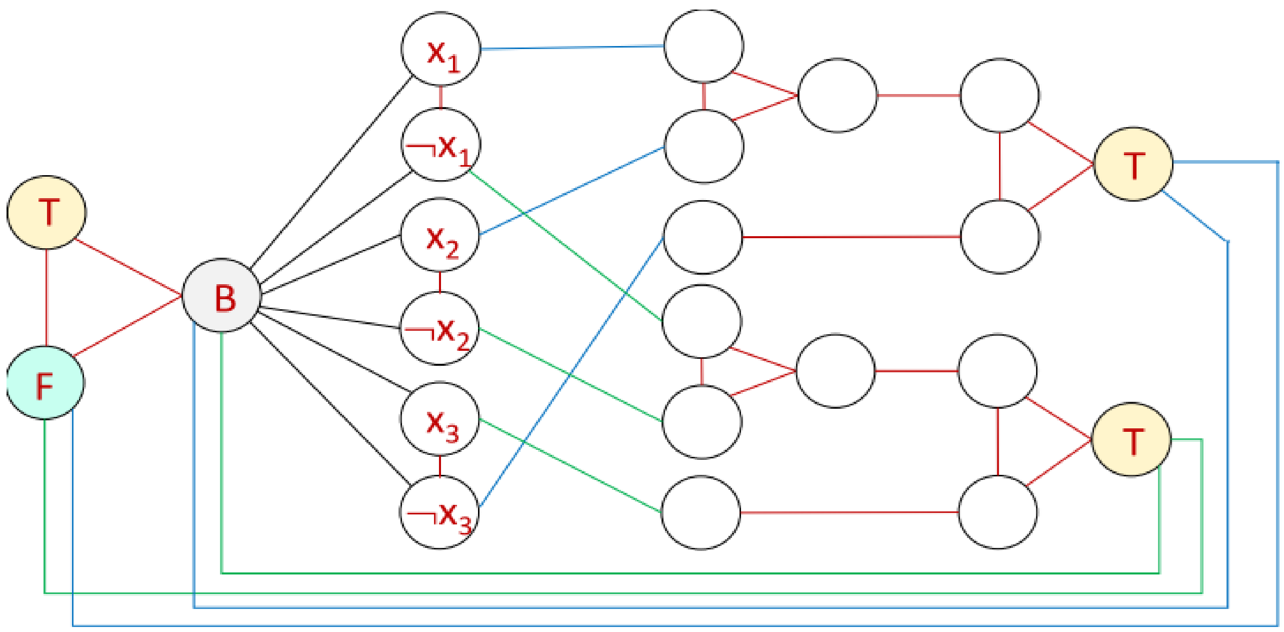


Coloring

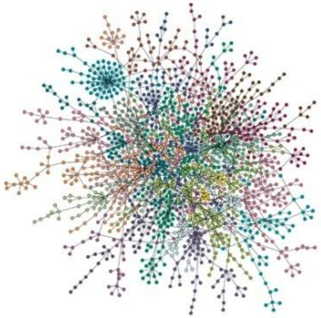


- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT $\leq$ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

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

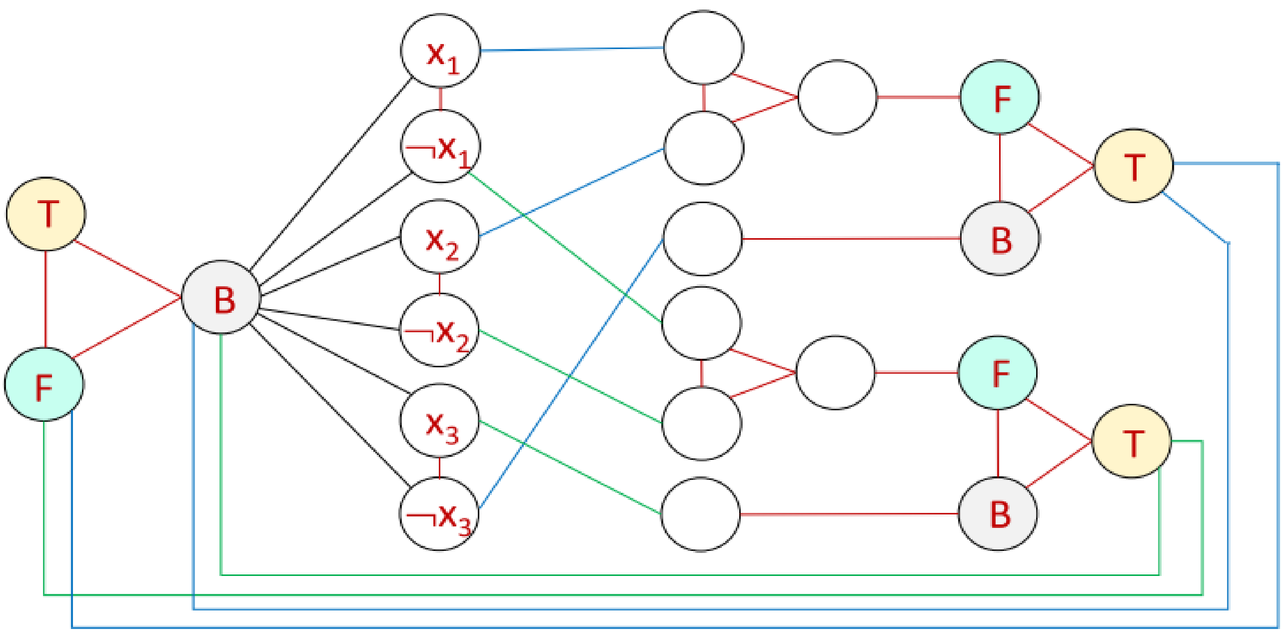


COLORING



- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT $\leq$ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

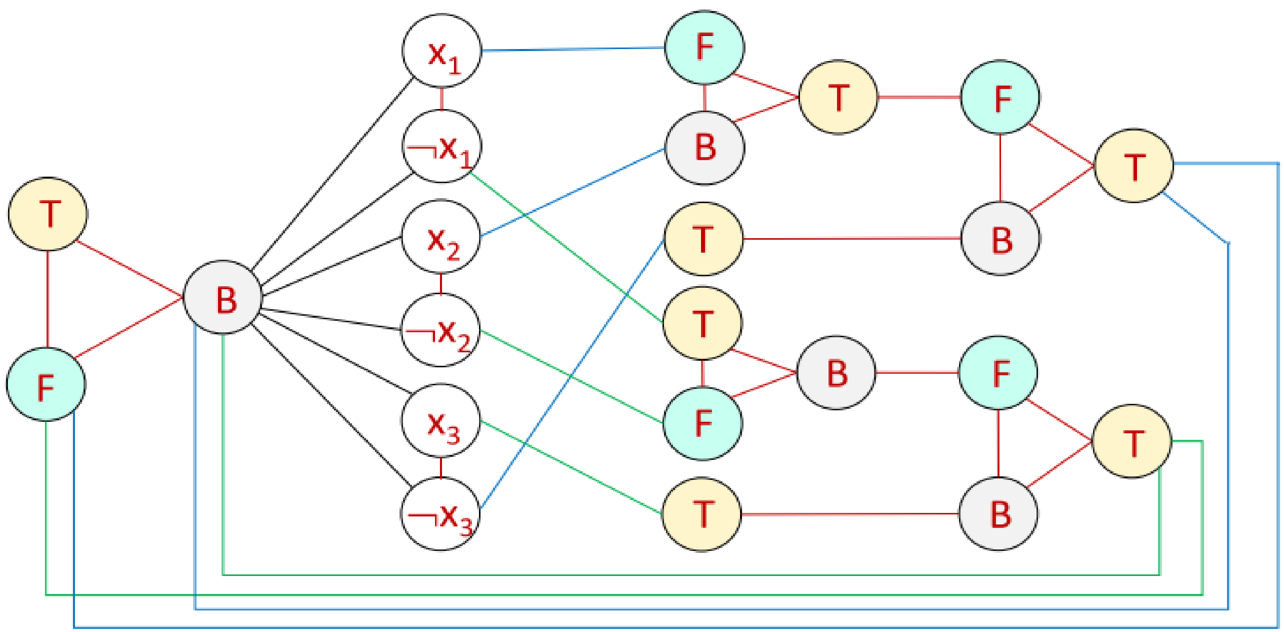
$$\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)$$



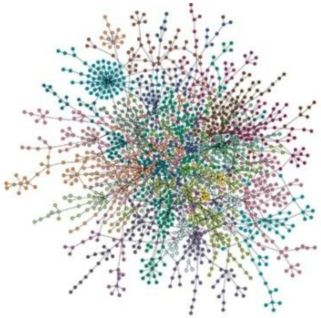
Coloring

- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT ≤ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$

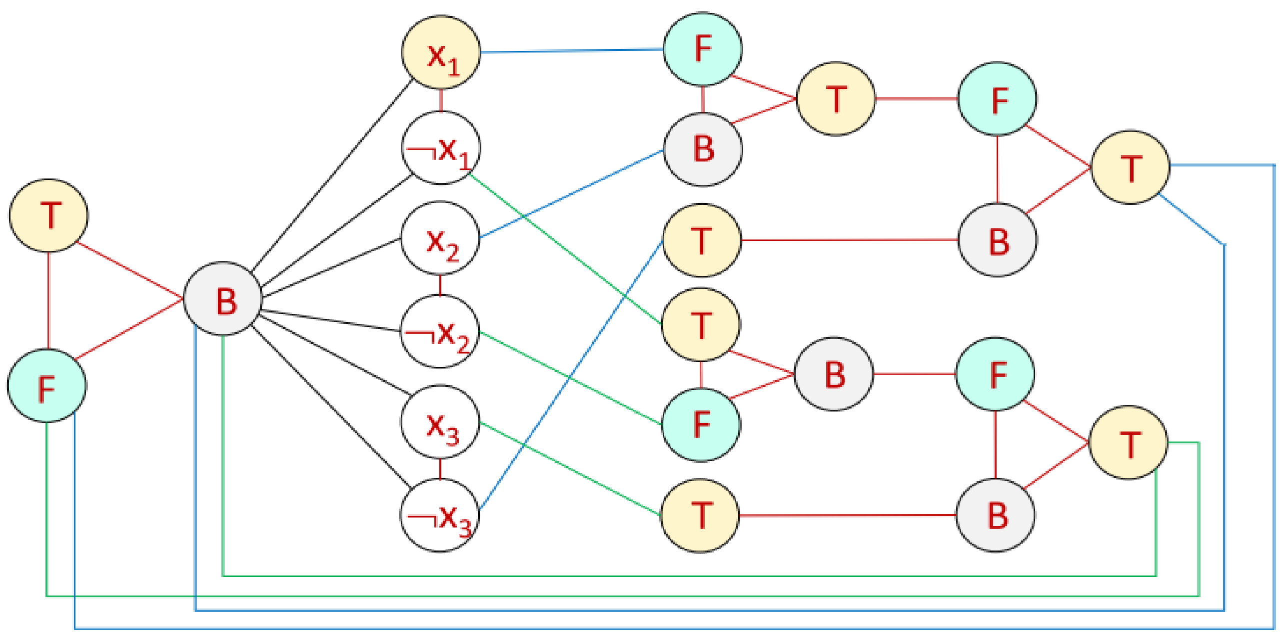


COLORING

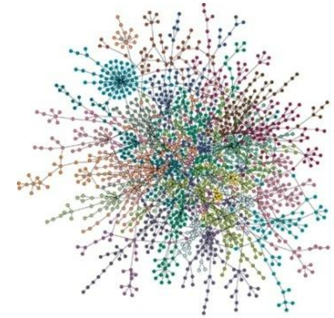


- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT ≤ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$



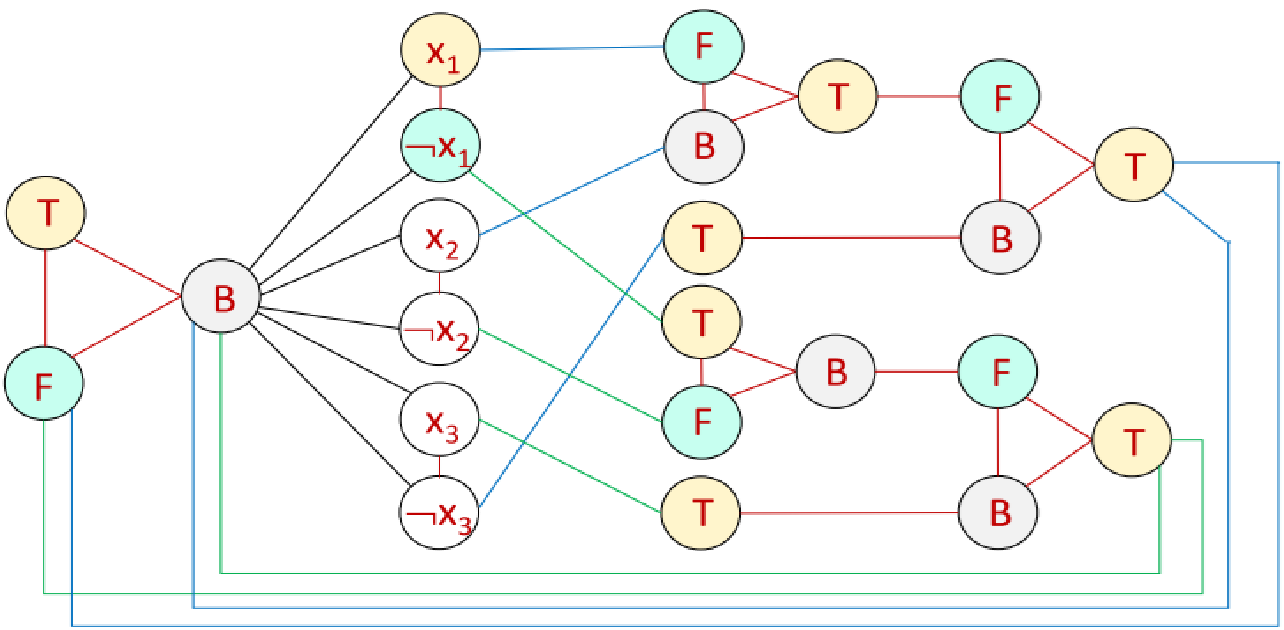
Coloring



3-Coloring Problem is NP-Complete

- **We can use the 3-Coloring problem to solve the 3SAT problem**
- **We have shown that 3SAT ≤ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$

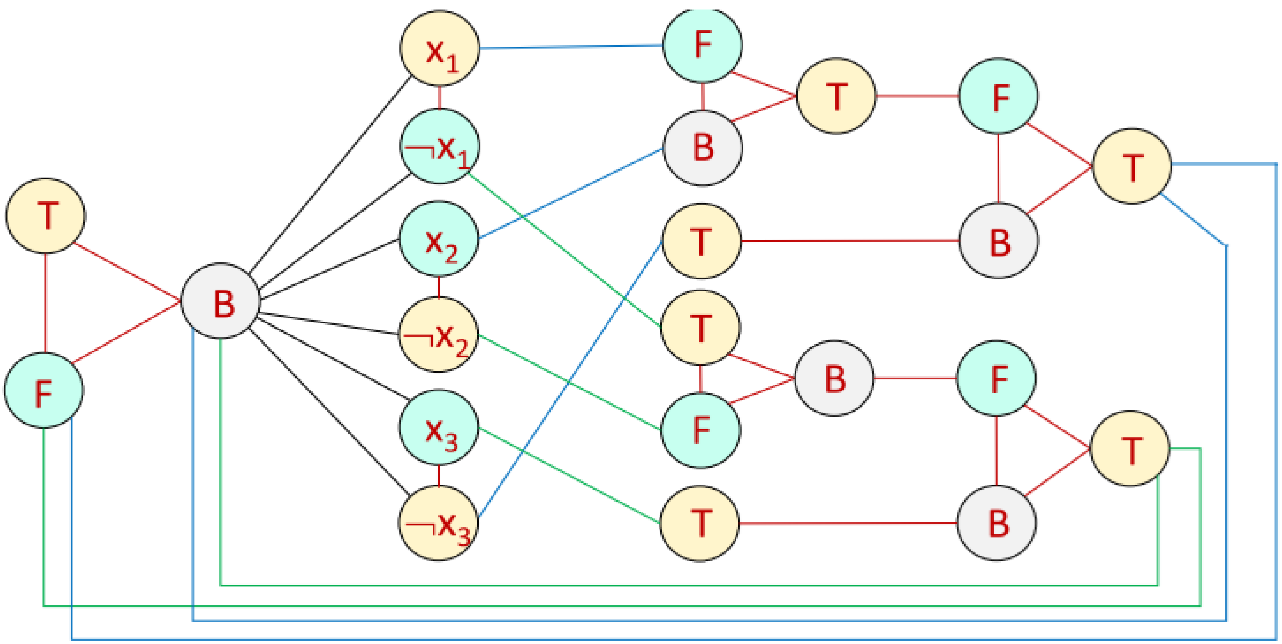


Coloring

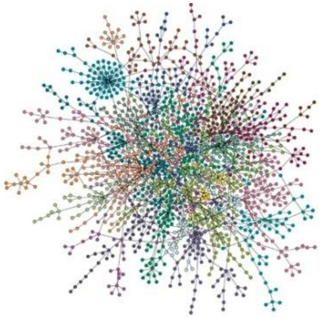
3-Coloring Problem is NP-Complete

- **We can use the 3-Coloring problem to solve the 3SAT problem**
- **We have shown that 3SAT ≤ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$

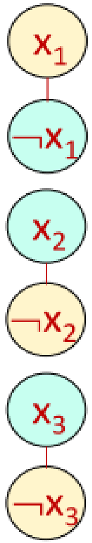


Coloring



- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT ≤ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

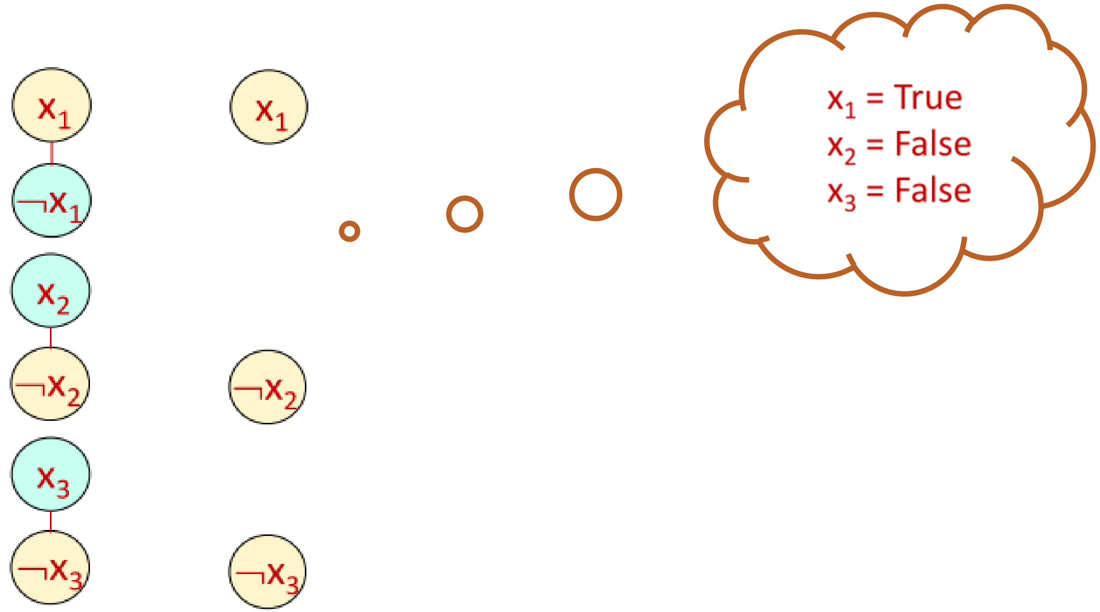
$$\boxed{\varphi = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_1)}$$



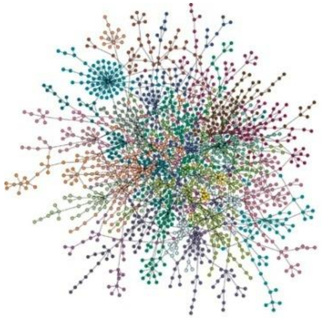
Coloring

- **3-Coloring Problem is NP-Complete**
  - **We can use the 3-Coloring problem to solve the 3SAT problem**
  - **We have shown that 3SAT ≤ 3-Coloring:** If we had a polynomial algorithm for the 3-Coloring problem we could solve the 3SAT problem in polynomial time.

$$\boxed{\varphi = \overset{\text{TRUE}}{(x_1 \vee x_2 \vee \neg x_3)} \wedge \overset{\text{TRUE}}{(\neg x_2 \vee x_3 \vee \neg x_1)}}$$

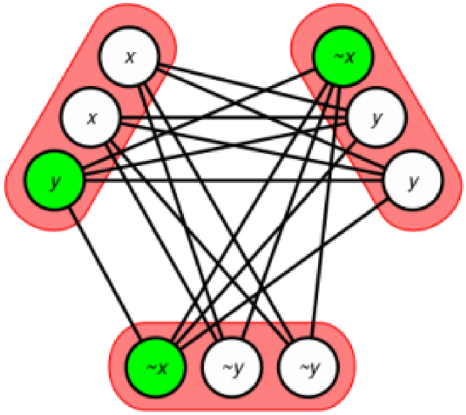


COLORING



3-Coloring Problem is NP-Complete

- 3-SAT is one of 21 NP-complete problems Karp, and is used as a reference problem for proof of NP-completeness of other problems.
- This is done through polynomial-time-reference from 3-SAT to other problems.
- An example is the clique problem: Given a snapshot (Boolean formula) ϕ of 3SAT with k terms $C_1, C_2, \dots, C_k$ (clauses) and n variables $x_1, x_2, \dots, x_n$ (variables), the corresponding graph consists of one node for each variable and an edge between two non-contradictory variables of different types.



- The graph has a k-clique if the logical type is satisfiable.