Deciding the On-line Chromatic Number of a Graph with Pre-Coloring is PSPACE-Complete

Christian Kudahl

University of Southern Denmark

kudahl@imada.sdu.dk

July 7, 2014

On-line graph coloring

- The vertices of a graph are revealed one by one.

On-line graph coloring

- The vertices of a graph are revealed one by one.
- When vertex, v, is revealed, it is revealed which vertices it is adjacent to (among those that were previously revealed).

On-line graph coloring

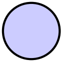
- The vertices of a graph are revealed one by one.
- When vertex, v, is revealed, it is revealed which vertices it is adjacent to (among those that were previously revealed).
- The algorithm assigns a color to v different from the colors it is already adjacent to. The color cannot be changed later.

On-line graph coloring

- The vertices of a graph are revealed one by one.
- When vertex, v, is revealed, it is revealed which vertices it is adjacent to (among those that were previously revealed).
- The algorithm assigns a color to v different from the colors it is already adjacent to. The color cannot be changed later.
- The goal is to use as few colors as possible.

An example

An example



An example

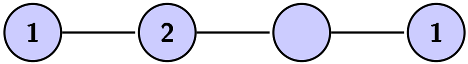
An example

An example

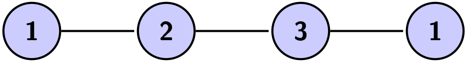
An example

An example

An example



An example



How to measure the quality of an algorithm?

Competitive Analysis [Sleator, Tarjan 85]

An algorithm A is c-competitive if there exists a b such that for any graph, G, the following holds:

$$A(G) \leq c \cdot \chi(G) + b$$

Here, $\chi(G)$ is the chromatic number of G. That is the minimum number of colors needed to color G (offline).

How to measure the quality of an algorithm?

Theorem [Gyarfas, Kiraly, Lehel 1990]

For any $k \in \mathbb{N}$, there exists a tree T_k such that any on-line algorithm can be forced to use k colors when on-line coloring T_k.

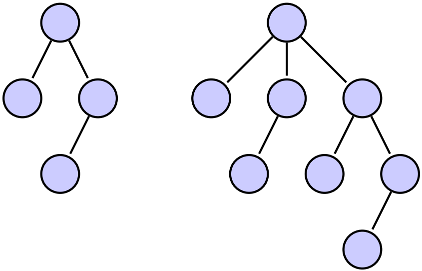


Figure: T_3 and T_4.

This means that no c-competitive algorithms exist for any constant c - even for the class of trees.

How to measure the quality of an algorithm?

On-line Competitive Analysis [Gyarfas, Kiraly, Lehel]

An algorithm A is on-line c-competitive if there exists an b such that for any graph, G, the following holds:

$$A(G) \leq c \cdot \chi^O(G) + b$$

Here, $\chi^O(G)$ is the on-line chromatic number of G. This is the smallest number such that there exists an algorithm that can color G using at most $\chi^O(G)$ colors for every ordering of the vertices.

On-line Chromatic Number

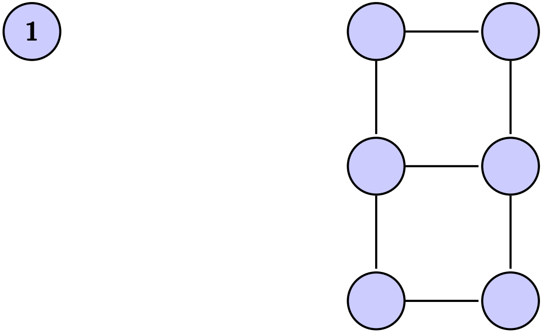


Figure: On-line coloring G. To the right, G is shown.

On-line Chromatic Number

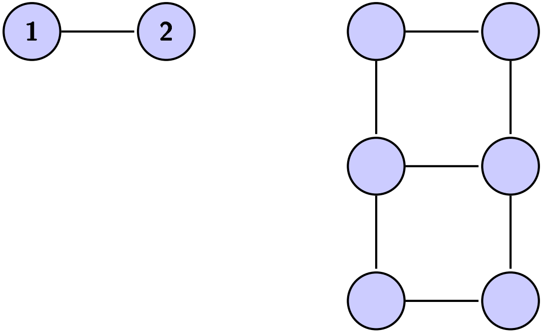


Figure: On-line coloring G. To the right, G is shown.

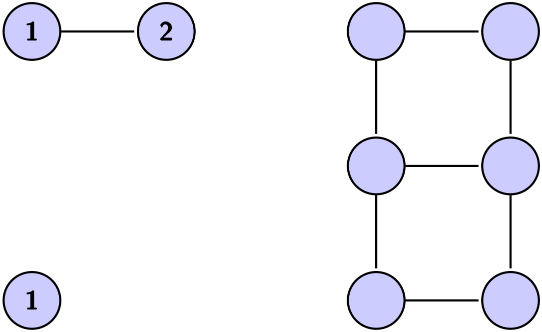


Figure: On-line coloring G. To the right, G is shown.

On-line Chromatic Number

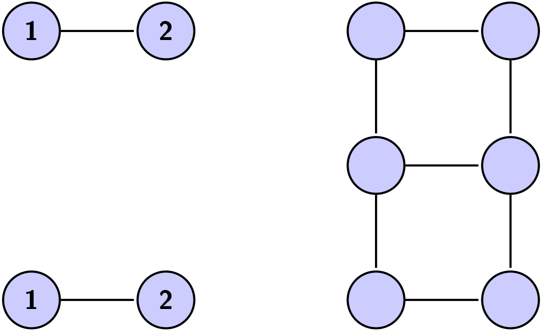


Figure: On-line coloring G. To the right, G is shown.

On-line Chromatic Number

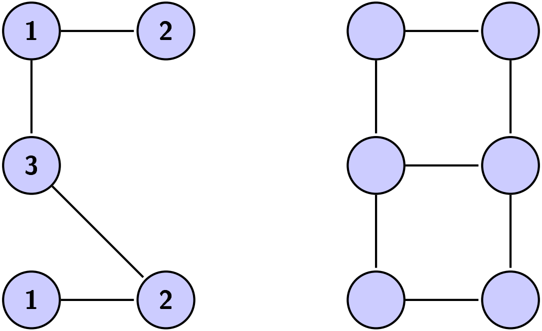


Figure: On-line coloring G. To the right, G is shown.

On-line Chromatic Number

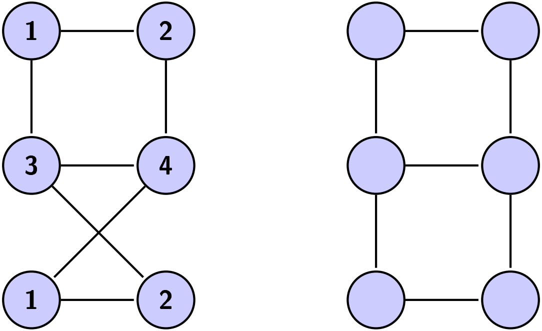


Figure: On-line coloring G. To the right, G is shown.

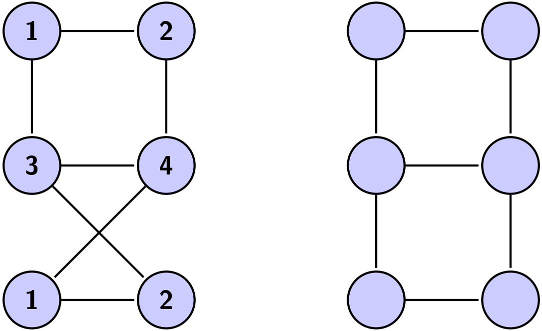


Figure: On-line coloring G. To the right, G is shown.

It turns out that $\chi^O(G) = 3$ (the bottom right vertex should have been given the color 3 instead).

On-line Chromatic Number

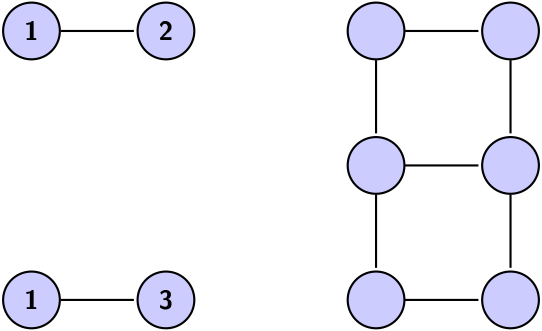


Figure: On-line coloring G. To the right, G is shown.

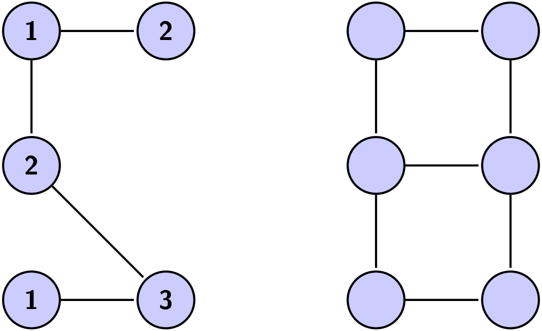


Figure: On-line coloring G. To the right, G is shown.

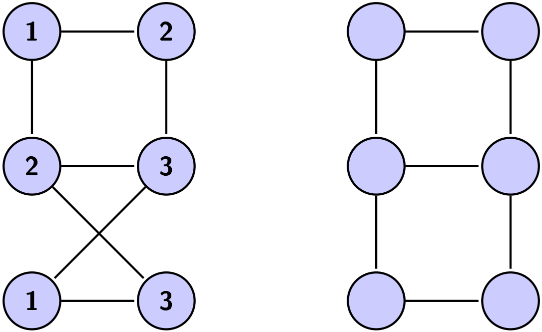


Figure: On-line coloring G. To the right, G is shown.

We know that it is NP-complete given a $k \in \mathbb{N}$ and a graph G to decide if $\chi(G) \leq k$.

Main Problem

How hard is it given a $k \in \mathbb{N}$ and a graph G to decide if $\chi^O(G) \leq k$?

On-line Graph coloring can be seen as a game between the Drawer and the Painter.

- (G, k) is the game instance they agree on before the game.

On-line Graph coloring can be seen as a game between the Drawer and the Painter.

- (G, k) is the game instance they agree on before the game.
- The Drawer presents a vertex on his turn.

On-line Graph coloring can be seen as a game between the Drawer and the Painter.

- (G, k) is the game instance they agree on before the game.
- The Drawer presents a vertex on his turn.
- The Painter colors that vertex on his turn.

On-line Graph coloring can be seen as a game between the Drawer and the Painter.

- (G, k) is the game instance they agree on before the game.
- The Drawer presents a vertex on his turn.
- The Painter colors that vertex on his turn.
- At each time, the presented graph is isomorphic to an induced subgraph of G.

On-line Graph coloring can be seen as a game between the Drawer and the Painter.

- (G, k) is the game instance they agree on before the game.
- The Drawer presents a vertex on his turn.
- The Painter colors that vertex on his turn.
- At each time, the presented graph is isomorphic to an induced subgraph of G.
- If the painter is able to color the entire graph without using more than k colors, he wins. Otherwise, he loses.

On-line Graph coloring can be seen as a game between the Drawer and the Painter.

- (G, k) is the game instance they agree on before the game.
- The Drawer presents a vertex on his turn.
- The Painter colors that vertex on his turn.
- At each time, the presented graph is isomorphic to an induced subgraph of G.
- If the painter is able to color the entire graph without using more than k colors, he wins. Otherwise, he loses.
- The painter has a winning strategy if and only if $\chi^O(G) \leq k$.

The On-line Chromatic Number

We can consider the following related problem:

On-line Chromatic Number with Pre-coloring

Given a state in the On-line Graph Coloring game, does the painter have a winning strategy from that state?

A state in the game means that a part of the graph has already been revealed and color.

This problem, which is also known as On-line Chromatic Number with Pre-coloring, is PSPACE-Complete.

Sketch of Proof

- Reduction from quantified satisfiability in 3-DNF.

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$

- Mapping F into graph G with precoloring and a k.

Sketch of Proof

- Reduction from quantified satisfiability in 3-DNF.

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$

- Mapping F into graph G with precoloring and a k.
- F true $\Leftrightarrow$ Painter can color G using at most k colors.

Sketch of Proof

- Reduction from quantified satisfiability in 3-DNF.

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$

- Mapping F into graph G with precoloring and a k.
- F true $\Leftrightarrow$ Painter can color G using at most k colors.
- F is false $\Leftrightarrow$ Drawer can force the painter to use more than k colors.

Sketch of Proof

- Reduction from quantified satisfiability in 3-DNF.

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$

- Mapping F into graph G with precoloring and a k.
- F true $\Leftrightarrow$ Painter can color G using at most k colors.
- F is false $\Leftrightarrow$ Drawer can force the painter to use more than k colors.
- Two vertices for each variable.

Sketch of Proof

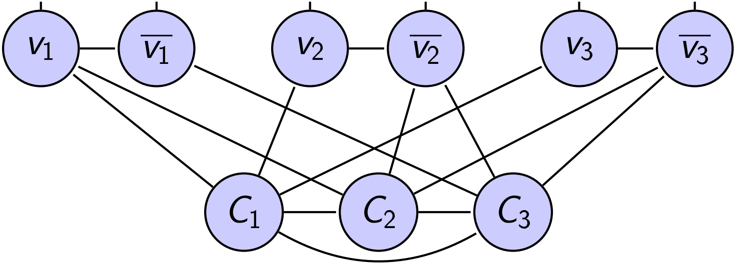
- Reduction from quantified satisfiability in 3-DNF.

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$

- Mapping F into graph G with precoloring and a k.
- F true $\Leftrightarrow$ Painter can color G using at most k colors.
- F is false $\Leftrightarrow$ Drawer can force the painter to use more than k colors.
- Two vertices for each variable.
- One vertex for each term.

Sketch of Proof

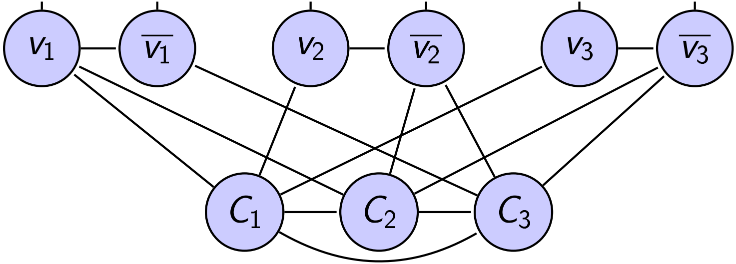
$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$



This is a part of the graph.

Sketch of Proof

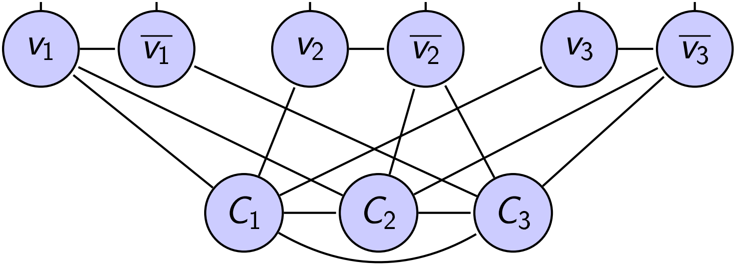
$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$



- Variables can only get color 'true' or 'false' (using pre-coloring)
- If C_i gets color 'false' (without conflict with variables), painter wins.
- Otherwise, he loses.

Sketch of Proof

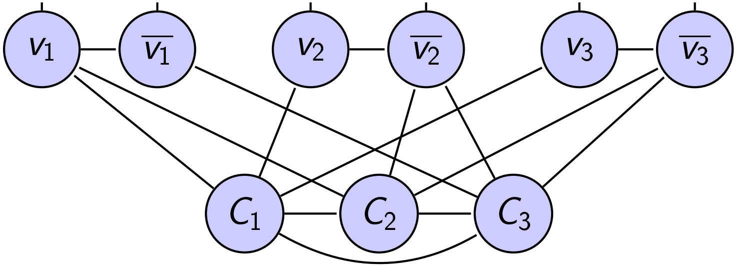
$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$



- When a variable or clause is requested, the painter knows the index (by pre-coloring).
- For the existentially quantified variables, the painter also know if it is v_i or $\overline{v_i}$.

Sketch of Proof

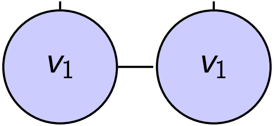
$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$



- If the formula is false, The Drawer can win.
- He requests the two vertices for each variable in order.
- For the universally quantified variables, he can decide the truth assignment.

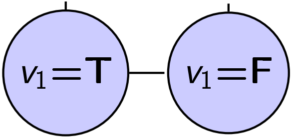
Sketch of Proof

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$



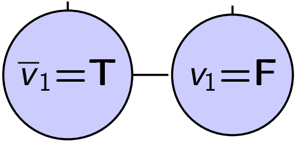
Sketch of Proof

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$



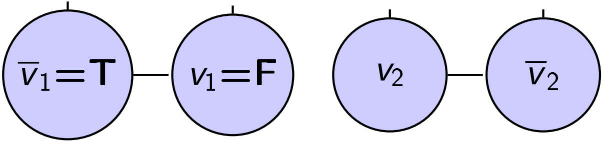
Sketch of Proof

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$



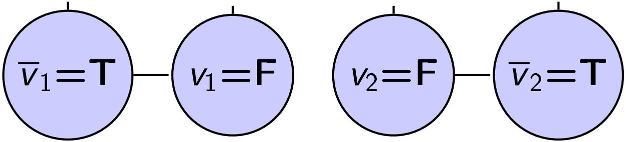
Sketch of Proof

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$



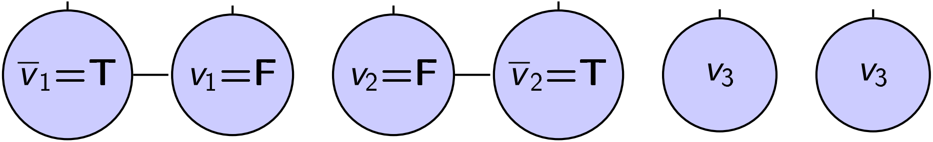
Sketch of Proof

$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$



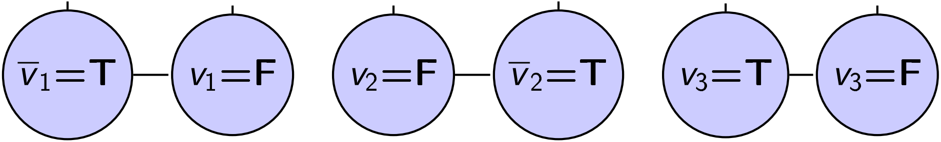
Sketch of Proof

$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$



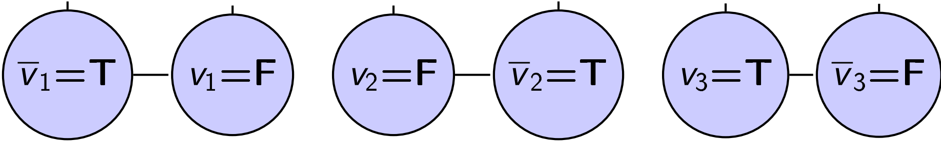
Sketch of Proof

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$



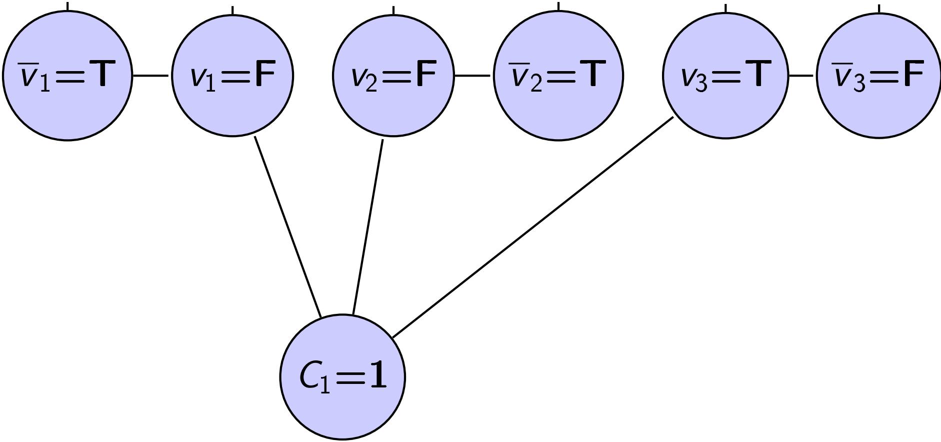
Sketch of Proof

$$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$$



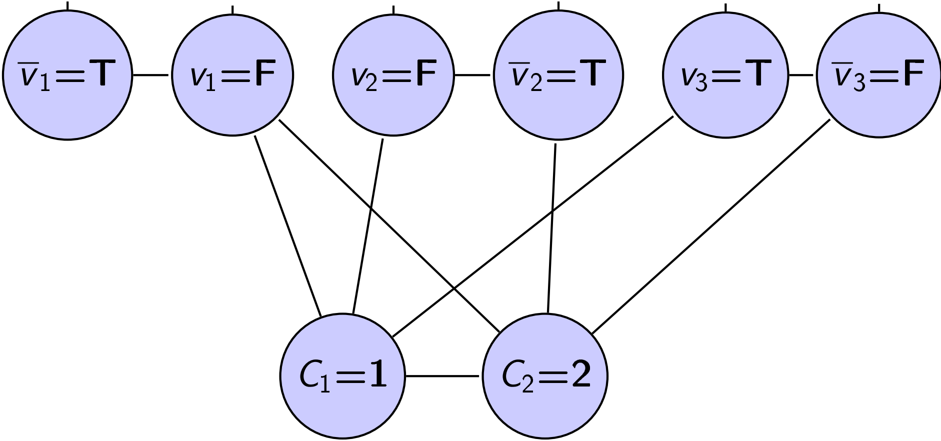
Sketch of Proof

$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$



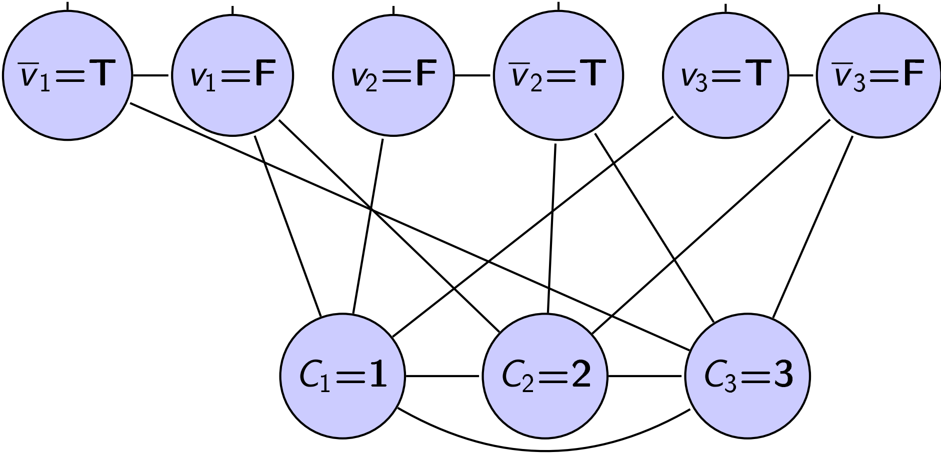
Sketch of Proof

$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$



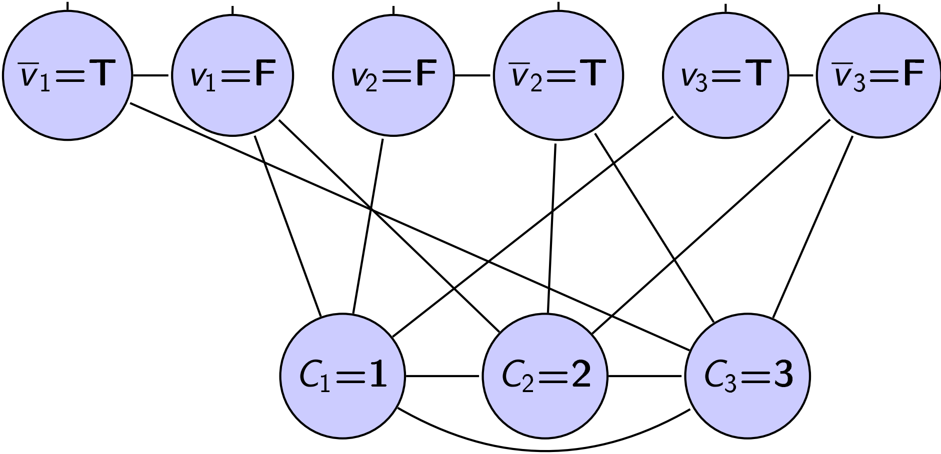
Sketch of Proof

$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$



Sketch of Proof

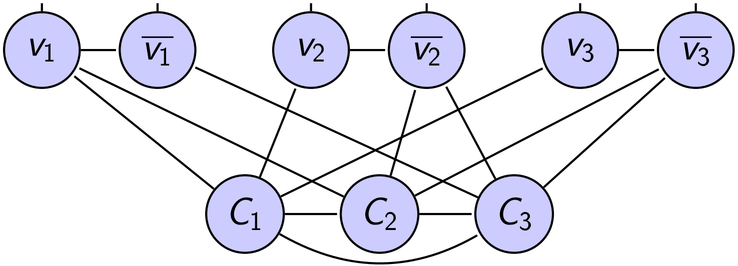
$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$



If one clause had been satisfied, it could have received the color false and the painter would have won.

Sketch of Proof

$F = \forall x_1 \exists x_2 \forall x_3 : (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge \overline{x}_2 \wedge \overline{x}_3) \vee (\overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3)$



- An additional gadget ensures that the Drawer must request variables in order.
- When F is true, the Painter can color existentially quantified variables according to the 'correct' truth assignment from the satisfiability problem.
- One clause will have no neighbour with color false, so it can get that color.

Other results, Open Problems, and Conjectures

It is still unknown if the problem remains PSPACE-Complete without the pre-coloring.

Theorem

Online Chromatic Number without Pre-coloring is coNP-Hard.

Theorem

Online Chromatic Number without Pre-coloring is Σ_2^p-Hard in multigraphs.

Conjecture [Gyarfas, Lehel, Kiraly 98]

Given a graph, G, it is NP-hard to decide if $\chi^O(G) \leq 4$

Thank you for listening!