

Coloring Algorithms.

(4TIN 9170)

François DROSS

- Lecture 6.

Flashback to the previous lecture:

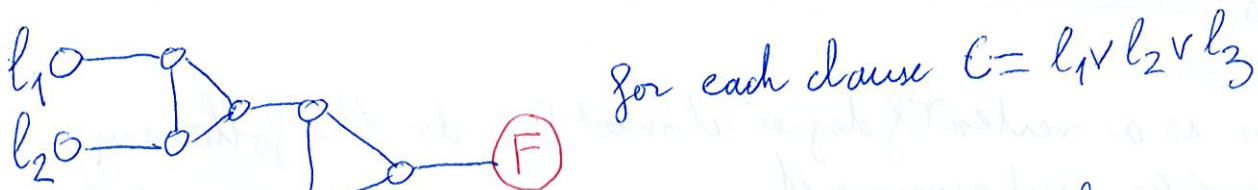
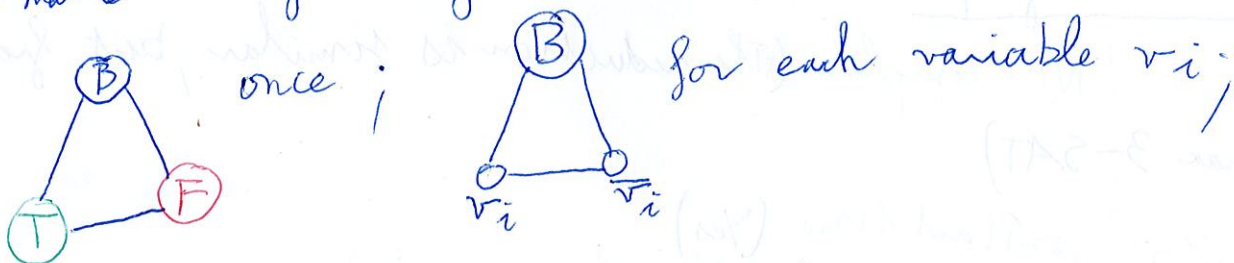
- (a) What does a 6-regular planar graph look like? *none exist*
- (b) What does a 5-regular planar graph look like? *if connected and faces are triangles 12 vertices*
- (c) What about vertices of small degree in ~~triangular~~-free planar graphs? *one vertex of degree ≤ 3*

- Complexity:

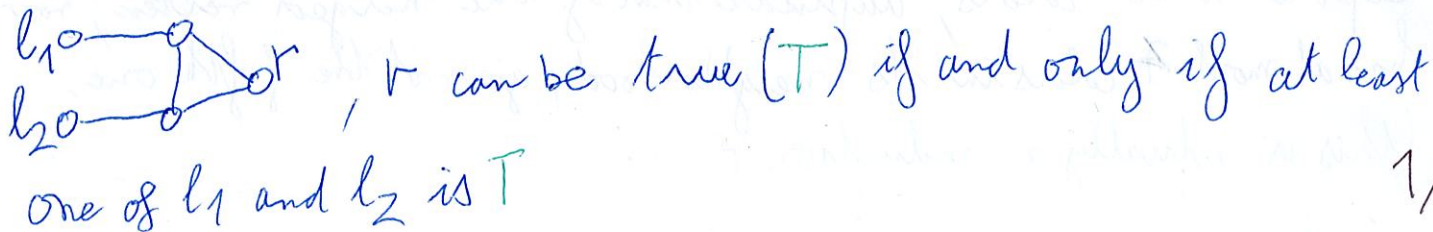
⚠ 3-coloring is NP-complete

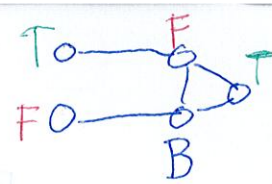
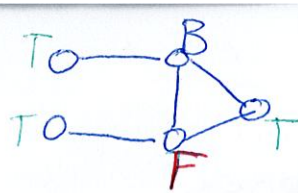
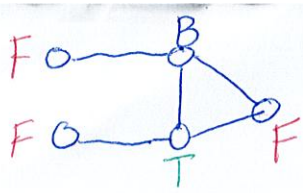
* proof: - given a coloring, one can check in time $O(n^2)$ if the coloring is proper \Rightarrow the problem is in NP.

- Let us make the following construction (reduction from 3-SAT).



- Assume there is a proper 3-coloring of the graph. Let B, T, and F be the 3 colors.
- Clearly for all i , either $v_i = F$ and $\bar{v}_i = T$, or $v_i = T$ and $\bar{v}_i = F$
- Now let us show that in:





- The first one is forced up to exchanging B and T, thus there is no coloring where l_1, l_2 and l_3 are F.
 - The existence of the last two implies that there is a coloring that respects each other possible constraint.
- ⇒ Therefore the coloring implies a solution to the 3-SAT formula.

Reciprocally, A solution to the 3-SAT formula implies a coloring of the graph.

⚠ What about ~~2~~ 2-coloring? It's polynomial! (just color greedily)

In planar graphs ⚠ 2 colors? polynomial.

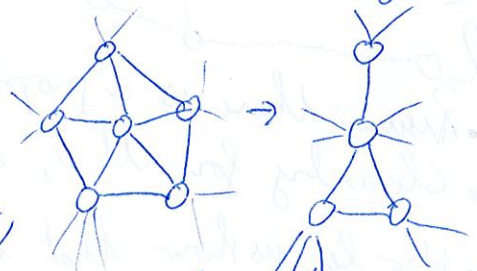
⚠ 3 colors? NP-complete (the reduction is similar, but from planar 3-SAT)

⚠ 4+ colors? constant time (Yes)
constructing a coloration (still on planar graphs)

⚠ 5 colors in $O(n^2)$:

⚠ While there is a vertex of degree at most 5: do the following:

- if degree ≤ 4 , just remove it
- if degree 5, merge two neighbors of v :



To get a coloring of the graph before the merge, keep the same colors, duplicate that of the merged vertex; now v has at most 4 colors in its neighborhood, give it the fifth one.
 (this is actually a reduction)

* Another proof that it can be done in $O(n^2)$: (still for 5 colors)

Take a planar graph; while there is a vertex of degree at most 5, remove it, color the rest of the graph recursively, do a Kempe change, and give the remaining color to v .

* 5 coloration in planar graphs in $O(n)$:

Keep track of the vertices of degree ≤ 5 in a list. Update the list when you remove a vertex, by checking its neighbors.

↳ preprocessing in $O(n)$, algorithm in $O(n)$.

And if you want 4 colors?

⚠ Algorithm in $O(n^2)$ (from the discharging proof).

* can we find a 4-coloration in $o(n^2)$? open question.

Lower bounds on the complexity.

⚠ complexité de lecture: on doit lire l'entrée entièrement.

* Attention cependant, par exemple calcul de matrice en moins de n^3 !

⚠ Exponential time hypothesis (ETH) (Rappel: ? $P \neq NP$; ? $W[1]$ -dur ? FPT)

The 3-SAT problem cannot be solved in $O(2^{\delta n})$, for a certain $\delta > 0$.

In other words there exists a $\delta > 0$ such that 3-SAT cannot be solved in $O(2^{\delta n})$.

* One can reduce certain problems, for instance 3-coloring, from 3-SAT, to show that if ETH holds, then they cannot be solved in $O(2^{\delta n})$ for a certain δ .

In those reductions, one must show that the size of the input does not change too much from that of the output.

* poly(n) · 2^n algorithm for k-coloring.

The naive algorithm runs in time poly(n) · k^n.

To show G is k-colorable $\Leftrightarrow \sum_{X \subseteq V(G)} (-1)^{|X|} g(X)^k \neq 0$

where $g(X) =$ nb of distinct stable sets in $G[X]$

$$\sum_{X \subseteq V(G)} (-1)^{|X|} g(X)^k \neq 0 \Leftrightarrow \sum_{X \subseteq V(G)} (-1)^{|X|} \left(\sum_{\substack{I_1, \dots, I_k \\ \text{stable sets in} \\ G[X]}} 1 \right) \neq 0$$

$$\Leftrightarrow \sum_{X \subseteq V(G)} \sum_{\substack{I_1, \dots, I_k \\ \text{stable sets} \\ \text{in } G[X]}} (-1)^{|X|} \neq 0$$

$$\Leftrightarrow \sum_{\substack{I_1, \dots, I_k \\ \text{stable sets in } G}} \sum_{X \supseteq I_1, \dots, I_k} (-1)^{|X|} \neq 0$$

Lemma: Set X ; $\sum_{A \subseteq X} (-1)^{|A|} = \begin{cases} 0 & \text{if } X \neq \emptyset \\ 1 & \text{if } X = \emptyset \end{cases}$

Proof: • if $X = \emptyset$, trivial

• if $X \neq \emptyset$: $\exists u \in X$

$$\sum_{A \subseteq X} (-1)^{|A|} = \sum_{A \subseteq X \setminus \{u\}} (-1)^{|A|} + \sum_{A \subseteq X \setminus \{u\}} (-1)^{|A|+1} = 0$$

• Here, $Y = V \setminus \{I_1 \cup \dots \cup I_k\}$

$$\sum_{A \subseteq Y} (-1)^{|A| + |I_1 \cup \dots \cup I_k|} = 0 \text{ except if } Y = \emptyset, \text{ i.e. } I_1 \cup \dots \cup I_k \text{ covers } V(G)$$

In that case, there is a coloring of G. If $Y \neq \emptyset$, $\sum_{X \supseteq I_1, \dots, I_k} (-1)^{|X|} = (-1)^{|I_1 \cup \dots \cup I_k|} \cdot 1$
(if $Y = \emptyset$) $= (-1)^{|V(G)|}$

Hence if $Y \neq \emptyset$, $\sum_{\substack{I_1, \dots, I_k \\ \text{stable sets in } G}} \sum_{X \supseteq I_1, \dots, I_k} (-1)^{|X|} \neq 0$ (it has the sign of $(-1)^{|V(G)|}$)

Now the algorithm is to compute all of the $g(x)$ for all $x \in V(G)$ by dynamic programming. \rightarrow time $\text{poly}(n) \cdot 2^n$.

Approximation algorithms:

* On 3-colorable graphs: $O(n^{1/2})$ colors in $\text{poly}(n)$ time.

but if one can do a constant number of colors in $\text{poly}(n)$ time, then some hypothesis (not $P \neq NP$, nor ETH. \dots) fails.

! Quasi-polynomial algorithms: 3-coloration on planar graphs in time $O(2^{\sqrt{n}})$

* Let $\epsilon > 0$ 3 coloring in $\text{poly}(n)$ time on graphs with min degree $\geq \epsilon n$:

- dominating set of size $\leq c \log n$ (where c varies with ϵ)

- Test all of the colorings of the dominating set \rightarrow a list of 2 colors remain on every vertex. \Rightarrow $\text{poly}(n)$ time \rightarrow List coloring (Nov 26th)

choices for next time:

- FPT - discharging 2. 3
- Tree width 3
- χ -bounded graph classes. 3