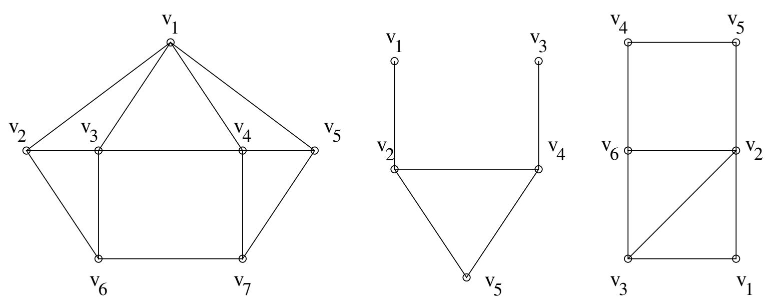
Complexité du coloriage de graphes

Correction.

NB: l'unique question importante de cet énoncé est la question 7. Elle sera la question qui aura le plus fort coefficient, de loin.

Dans tout le problème, un *graphe* désigne un graphe non orienté, c'est-à-dire un couple $G = (V, E)$, où V est un ensemble fini de *sommets*, et E est un ensemble de paires $\{v_1, v_2\}$ de sommets appelées *arêtes*. Si $\{v_1, v_2\} \in E$, on dit que v_1 et v_2 sont *adjacents*, et qu'ils sont les *extrémités* de l'arête $\{v_1, v_2\}$. Le *degré* $d_G(v)$ d'un sommet v de G est le nombre de sommets adjacents à v dans G. Le *degré* $d(G)$ du graphe G est $\max_{v \in V} d_G(v)$.

On notera souvent les graphes par des dessins, par exemple:



[1] [2] [3]

Pour tout entier $k \in \mathbb{N}$, un k-*coloriage* de G est une application c de V vers $[1, k]$ telle que $c(v_1) \neq c(v_2)$ pour tous sommets adjacents v_1, v_2. De façon imagée, on appelle $c(v)$ la *couleur* du sommet v de G dans c. Si G a un k-coloriage, on dit que G est k-*colorable*. Le plus petit entier k tel que G est k-colorable est le *nombre chromatique* de G.

Pour tout ensemble fini S, $|S|$ désigne le cardinal de S.

On rappelle que le problème 3-SAT de la satisfiabilité d'ensembles de 3-clauses est NP-complet. Un *littéral* L est soit une variable propositionnelle x soit une négation $\overline{x}$ d'une variable propositionnelle x. Une 3-*clause* est une disjonction de trois littéraux, non nécessairement distincts. Un *ensemble de* 3-*clauses* S est vu comme une conjonction de ses 3-clauses. Une *valuation* ρ est une application qui à chaque variable x associe un booléen, VRAI ou FAUX; ρ *satisfait* une 3-clause si elle contient x tel que $\rho(x) =$VRAI ou $\overline{x}$ tel que $\rho(x) =$FAUX; ρ *satisfait* un ensemble S si ρ satisfait toute 3-clause de S. S est *satisfiable* si et seulement si S est satisfait par au moins une valuation ρ.

1. Soit G un graphe 2-colorable. Soit n la couleur du sommet v. Quelles sont les couleurs des sommets adjacents à v ?

    Les couleurs des sommets adjacents à v sont différentes de n, mais il n'y a que deux couleurs disponibles en tout, donc ils ont tous la même couleur, à savoir $3 - n$, ou encore 2 si $n = 1$ et 1 si $n = 2$.

2. Montrer que G est k-colorable si et seulement si toutes ses composantes connexes sont k-colorables. Une *composante connexe* de G est un sous-graphe maximal de G dans lequel il existe un chemin entre v et v' pour tous sommets v et v'.

    Si G est k-colorable, alors il est clair que toute composante connexe de G l'est aussi.

    Réciproquement, supposons que toutes les composantes connexes $G_1, \ldots, G_n$ de G sont k-colorables, et soient $c_1, \ldots, c_n$ respectivement des k-coloriages de ces composantes connexes. Les domaines de définition de $c_1, \ldots, c_n$ sont disjoints, considérons donc l'application $c = c_1 \uplus \ldots \uplus c_n$. Il s'agit d'un k-coloriage. En effet, si v et v' sont deux sommets adjacents, ils sont dans la même composante connexe par définition, disons G_i, et $c[v] = c_i[v] \neq c_i[v'] = c[v']$.

3. Déduire des questions précédentes que le problème de la 2-colorabilité d'un graphe G donné en entrée est décidable en temps polynomial.

    On n'a pas dit jusqu'ici comment le graphe G était censé être présenté sur la bande d'entrée. G peut être présenté par matrice d'adjacence, par listes d'adjacences, par exemple. Pourquoi le résultat de décidabilité en temps polynomial du 2-coloriage est-il indépendant du choix de la représentation de G parmi celles-ci?

    Supposons G 2-colorable. Choisissons un sommet quelconque de G. On peut supposer sans perte de généralité (échanger les couleurs 1 et 2 au besoin), qu'il est colorié 1. Alors tous les sommets adjacents sont coloriés 2, les sommets adjacents à ces derniers sont coloriés 1, et ainsi de suite. En somme, si G est connexe, G n'a qu'un 2-coloriage, à échange de couleurs près. Si G n'est pas connexe, notons que G est 2-coloriable si et seulement si toutes ses composantes connexes le sont. Ceci mène à l'algorithme :

```
fonction 2COL (V, E) {
        pour i = 1..|V| { c[i] := 0; } (* initialisation. *)
        tant que VRAI { (* on explore les composantes connexes une par une. *)
                i := 1; tant que i ≤ |V| et c[i] ≠ 0 { i := i + 1; }
                si i > |V| (* si pas de composante connexe restante, *)
                   alors retourner c; (* alors on a trouvé. *)
                (* sinon, on passe à la composante connexe suivante. *)
                si 2COL_CONNEXE (V, E, c, i, 1) =FAUX
```

```
                  alors retourner NON-2-COLORIABLE;
            }
        }

fonction 2COL_CONNEXE (V, E, c, i, couleur) (* colorie i par couleur, et récurse. *)
        si c[i] = 0 (* si i n'est pas encore colorié, on le colorie, et on récurse. *)
          alors { c[i] := couleur;
                pour tout j ∈ V[i] {
                      si 2COL_CONNEXE (V, E, c, j, 3 − couleur) =FAUX
                        alors retourner FAUX;
                      }
                }
        sinon retourner c[i] = couleur;
        }
```

On peut représenter G par matrice d'adjacence ou listes d'adjacences: l'algorithme ci-dessus fonctionne par listes d'adjacence, mais la conversion depuis les matrices d'adjances s'effectue en espace logarithmique, comme il a été vu en TD, donc en temps polynomial.

4. Montrer que le graphe [2] a la propriété suivante : dans tout 3-coloriage de ce graphe, si v_1 et v_3 sont de couleurs autres que 3, et si v_5 est de couleur 1, alors v_1 ou v_3 est de couleur 1. (Le graphe [2], intuitivement, code donc une disjonction.)

   Pour montrer que v_1 ou v_3 est de couleur 1, sachant qu'ils ne sont pas de couleur 3, il suffit de montrer que v_1 et v_3 ne peuvent pas être tous les deux de couleur 2. En effet, si cela était le cas, alors v_2 et v_4 seraient de couleurs différentes de 2, et différentes entre elles, c'est-à-dire de couleurs 1 et 3, ou 3 et 1. Comme v_5 est de couleur différente de ces deux dernières, il serait de couleur 2, ce qui contredit le fait qu'il est de couleur 1.

5. Trouver un graphe ayant au moins deux sommets v_1 et v_2, tel que dans tout 3-coloriage, si v_1 et v_2 sont de couleurs différentes de 3, alors v_2 est de couleur 1 si et seulement si v_1 n'est pas de couleur 1. (Ceci code une négation.)

   Le graphe ayant juste v_1 et v_2 comme sommets et exactement une arête entre les deux convient clairement, puisqu'ils ne peuvent avoir que les couleurs 1 ou 2 par hypothèse.

6. Si un graphe a deux sommets A de couleur 3 et B de couleur 2, dans un 3-coloriage donné, quelles sont les couleurs des sommets adjacents à la fois à A et à B ? (Si on code les formules logiques par des sommets, et que "être de couleur 1" signifie "être vrai", ceci code une conjonction.)

Trivial : il ne reste qu'une couleur possible, 1.

7. Déduire des trois questions précédentes qu'il existe un algorithme en temps polynomial qui transforme tout ensemble S de 3-clauses en un graphe G qui est 3-colorable si et seulement si S est satisfiable. (Indication : on créera deux sommets distingués A et B dans G, adjacents; par symétrie, on supposera que A est de couleur 3 et B de couleur 2, et on connectera les sous-graphes des questions précédentes à l'un ou l'autre de ces sommets ou même aux deux.)

    Intuitivement, il faut pouvoir coder la disjonction et la négation, et le fait d'être VRAI*. Utilisons pas convention la couleur* 1 *pour représenter* VRAI*, et* 2 *pour représenter* FAUX*. D'après la question 4, le graphe [2] code la disjonction, à ceci près qu'il faut pouvoir garantir que les entrées* v_1 *et* v_3 *sont de couleurs différentes de* 3*. Pour cela, on crée un sommet spécial* A *dans le graphe* G *à venir. Par symétrie, on pourra toujours supposer qu'il sera de couleur* 3*. Pour assurer que* v_1 *et* v_3 *sont de couleurs différentes de* 3*, il suffira donc de poser des arêtes reliant* A *à* v_1 *et* v_3*.*

    Pour coder la négation, de même, il suffit d'exprimer que la négation d'un sommet est un sommet qui lui est relié par une arête, et qui est aussi relié à A*, par la question 5.*

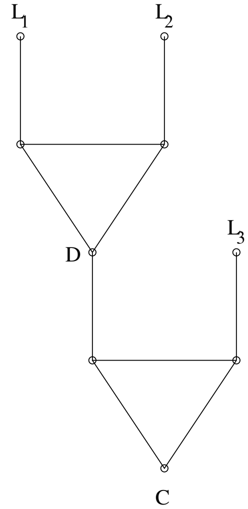
    Pour coder le fait qu'une série de sommets, chacun représentant une clause, doivent tous être coloriés 1*, on ne peut pas se contenter de les relier à* A*, ce qui imposerait seulement qu'ils seraient de couleurs* 1 *ou* 2*. Par contre, on peut créer un nouveau sommet* B*, relié à* A*, et qui par symétrie pourra être supposé de couleur* 2*. Si on relie tous les sommets représentant les clauses à la fois à* A *et à* B*, ceci imposera que tous ces sommets seront coloriés* 1*, par la question 6.*

    Définissons formellement cette construction. Pour tout ensemble S *de* 3*-clauses, on construit un graphe* G *comme suit :*

    (a) D'abord, G *contient deux sommets distingués* A *et* B*, et une arête entre* A *et* B*.*

    (b) Ensuite, pour toute variable x *dans* S*, on crée deux sommets* x *et* $\overline{x}$*, une arête entre* x *et* $\overline{x}$*, et deux arêtes reliant* A *à* x *et à* $\overline{x}$*. (On a ainsi naturellement, pour chaque littéral* L*, un sommet correspondant à* L*.)*

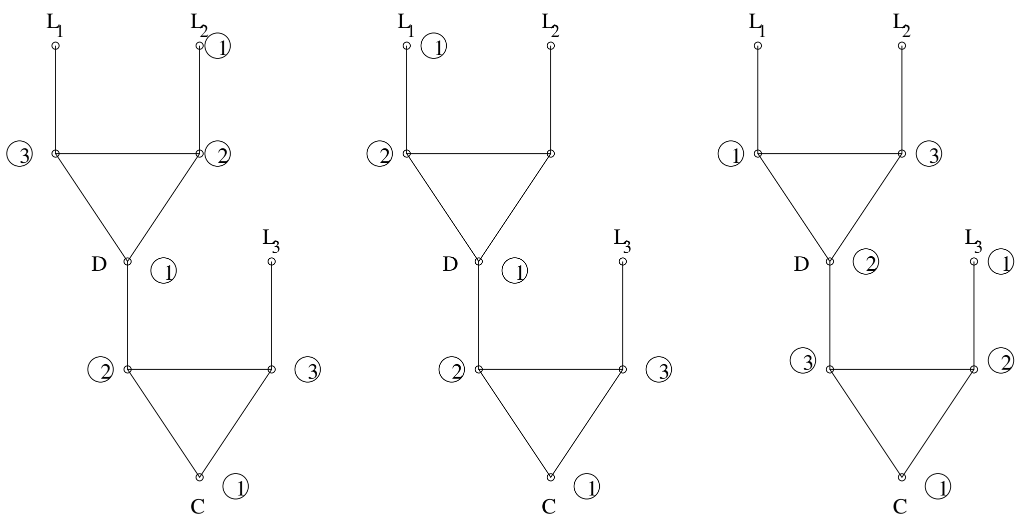
    (c) Pour chaque clause $C = L_1 \vee L_2 \vee L_3$ *dans* S*, on crée un sommet dénoté* C*, plus 5 autres sommets en combinant deux graphes de la forme [2] :*



[4]

(d) *On relie finalement tous les sommets C ainsi obtenus à l'étape précédente à la fois à A et à B, et les sommets D à A.*

Montrons que si S est satisfiable, alors G est 3-coloriable. Soit donc ρ une valuation satisfaisant S, et assimilons 1 à VRAI*, 2 à* FAUX*. Colorions chaque sommet x par $\rho(x)$, et $\overline{x}$ par la couleur correspondant à la négation de $\rho(x)$. On colorie A par 3, B par 2, ce qui colorie correctement les arêtes des points 7a et 7b de la construction. Noter que pour toute clause $C = L_1 \vee L_2 \vee L_3$ de S, au moins un des littéraux L_1, L_2 ou L_3, vu en tant que sommet, a reçu la couleur 1. Colorions chaque graphe [4] par le graphe de gauche si L_1 est colorié 1, par celui du milieu si L_2 est colorié 1, par celui de droite si L_1 n'est pas colorié 1 mais L_3 l'est :*



Il s'ensuit que les arêtes du point 7c de la construction sont bien coloriées. Comme tous les sommets C sont coloriés 1, et tous les sommets D sont coloriés 1 ou 2, les arêtes du point 7d sont aussi bien coloriées.

Réciproquement, supposons G 3-colorié. Sans perte de généralité, on peut supposer que A est colorié 3, et B, qui doit avoir une couleur différente de A par le *point 7a de la construction, est colorié* 2. *Par le point 7b, les sommets* x *et* $\overline{x}$ *sont de couleurs* 1 *ou* 2, *et de couleurs différentes. Notons* ρ *la valuation qui à* x *associe* VRAI *si* x *est de couleur* 1, FAUX *si* x *est de couleur* 2. *Mais, pour toute clause* $C = L_1 \vee L_2 \vee L_3$, *par le point 7d,* C *est de couleur* 1. *Ceci implique que* D *ou* L_3 *est colorié* 1; *en effet la question 4 s'applique, parce que* D *et* L_3 *sont de couleurs différentes de 3 : dans le cas de* D, *c'est parce que* D *est relié à* A *par le point 7c, et pour* L_3 *c'est parce que c'est un* x *ou un* $\overline{x}$, *qui est relié à* A *par le point 7b. De plus, si* D *est colorié* 1, *alors par le point 7b et la question 4 encore,* L_1 *ou* L_2 *est colorié* 1. *Donc au moins un des* L_i, $1 \leq i \leq 3$, *est colorié* 1. *Mais par définition de* ρ, L_i *est colorié* 1 *si et seulement si* L_i *est* VRAI *dans* ρ. *Donc* ρ *satisfait toutes les clauses* C *de* S, *donc* S *est satisfiable.*

8. Justifier informellement que l'algorithme de la question précédente s'exécute en espace logarithmique, moyennant quelques modifications éventuelles.

   Les étapes (7a) et (7c) s'effectuent facilement au fur et à mesure de la lecture de l'ensemble des clauses de S, *en n'utilisant qu'un espace de travail logarithmique. L'étape (7d) peut de plus être combinée avec l'étape (7c), en reliant le sommet* C *ainsi créé à* A *et à* B, *et en reliant* D *à* A *au fur et à mesure de l'examen de chaque clause* C.

   La seule difficulté réside dans l'étape (7b). On ne peut pas créer la liste des variables présentes dans S, *ce qui prendra un espace linéaire et non logarithmique. En revanche, on peut par exemple décider de donner des noms conventionnels aux nœuds créés lors de l'étape (7b). Si dans l'entrée la variable* x *est représentée par une chaîne de caractères, on pourra décider que le nom du sommet* x *est* $x00$ *et celui de* $\overline{x}$ *est* $x10$, *et que le nom de tous les autres sommets se terminera par* 1. *A l'étape (7c), on reprend pour* L_1, L_2, L_3 *les noms correspondants.*

9. En déduire que le problème de la 3-colorabilité des graphes est NP-complet.

   Par les questions précédentes, ce problème est NP*-difficile. Il ne reste plus qu'à observer qu'il est dans* NP*: étant donné un graphe, deviner les couleurs de chaque sommet parmi* $1, 2, 3$ *(ceci demande un nombre polynomial d'instructions* GUESS *d'une machine non déterministe), puis vérifier qu'il s'agit bien d'un coloriage en temps polynomial: parcourir les sommets du graphe, et pour chacun vérifier que tous les sommets adjacents ont des couleurs différentes.*