Partiel du 9 novembre 2011

Durée : 2 heures - Aucun document autorisé.

Exercice 1.
Machine à "shifter"

1. Décrire une machine de Turing déterministe à un ruban, d'alphabet d'entrée $\{a,b\}$, qui "shift" son entrée (*i.e.* qui remplace le mot $\mathrm{B}w_1 \ldots w_n\mathrm{B}\ldots$ initialement écrit sur son ruban par le mot $\mathrm{BB}w_1 \ldots w_n\mathrm{B}\ldots$).
☞ $\mathscr{M} = (Q, \Sigma, \Gamma, q_0, q_{\text{accept}}, q_{\text{rejet}}, \delta)$ où $\Sigma = \{a,b\}$; $\Gamma = \{a,b,\mathrm{B}\}$; $Q = \{q_0, q_a, q_b, q_{\text{accept}}, q_{\text{rejet}}\}$ et où δ est décrite par les instructions suivantes :

$$q_0.\mathrm{B} = (q_0, \mathrm{B}, \rightarrow) \quad ; \quad q_x.\mathrm{B} = (q_{\text{accept}}, x, \downarrow) \quad \text{pour } x,y \in \{a,b\}$$
$$q_0.x = (q_x, \mathrm{B}, \rightarrow) \quad ; \quad q_x.y = (q_y, x, \rightarrow)$$

2. Évaluer la complexité de cette machine.
☞ Il est facile de voir que $\mathscr{M}$ parcourt son entrée $w = w_1 \ldots w_n$ en substituant chaque lettre de ce mot comme indiqué par δ, puis effectue un dernier déplacement à droite pour remplacer le premier B de la configuration initiale par w_n avant de s'arrêter. Le calcul de $\mathscr{M}$ sur toute entrée de taille n consiste donc en une suite de $n+1$ configurations, autrement dit : $\mathscr{M}$ est de complexité en temps $n+1$.

Exercice 2.
Un peu de cours...

1. Définir la classe P.

2. Montrer que P est clos par complémentation (*i.e.* si un langage $L \subseteq \Sigma^*$ est dans P, alors son complémentaire $\Sigma^* \setminus L$ est dans P).

3. Définir la notion de réduction polynômiale.

Exercice 3.
2-SAT $\in$ P

Une *2-clause* est une clause comportant au plus deux littéraux (*e.g.*, $p \vee q$, $\neg p \vee q$, $\neg p$, etc). Montrez que le problème suivant est dans P :

2-SAT
Entrée : Une conjonction Φ de 2-clauses ;
Question : Φ est-elle satisfaisable ?

☞ À tout ensemble de 2-clause $\mathscr{C}$ on associe le graphe $G_{\mathscr{C}}$ défini comme suit :

$V_{G_{\mathscr{C}}}$ contient deux sommets x et $\neg x$ pour chaque variable x intevenant dans $\mathscr{C}$;

$E_{G_{\mathscr{C}}}$ contient un arc $\alpha \rightarrow \beta$ ssi la clause $\neg\alpha \vee \beta$ ou la clause $\beta \vee \neg\alpha$ est dans $\mathscr{C}$.

Alors $\mathscr{C}$ est satisfaisable ssi $G_{\mathscr{C}}$ ne comporte aucun circuit passant simultanément par un littéral et son opposé. Par ailleurs : la taille de $G_{\mathscr{C}}$ est polynomiale en celle de $\mathscr{C}$; le calcul de $G_{\mathscr{C}}$ à partir de $\mathscr{C}$ se fait en temps linéaire ; le test d'existence d'un chemin entre deux littéraux opposés dans $G_{\mathscr{C}}$ se fait en temps polynômial en $|G_{\mathscr{C}}|$ (et donc en $|\mathscr{C}|$). Le résultat en découle.

Exercice 4. $\mathsf{P} \neq \mathsf{NP}$ *?*

Trouvez l'erreur, dans cette preuve incorrecte de la conjecture « $\mathsf{P} \neq \mathsf{NP}$ » :

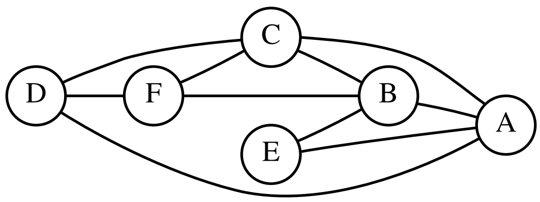
> *On considère l'algorithme suivant pour* SAT *: « Pour une formule propositionnelle φ prise en entrée, calculer $v(\varphi)$ pour toutes les valuations possibles v. Accepter si l'une des valuations envoie φ sur 1, refuser sinon. » Cet algorithme requiert clairement un temps exponentiel en la taille de φ. Ainsi,* SAT *est de complexité exponentielle, et donc n'appartient pas à* P. *Comme* SAT $\in \mathsf{NP}$, *il s'ensuit que* $\mathsf{NP} \neq \mathsf{P}$.

☞ L'existence d'un algorithme exponentiel pour SAT n'empêche en rien l'existence d'un algorithme polynomial pour ce problème. Donc on ne peut déduire de la complexité de l'algorithme précédent que SAT $\notin \mathsf{P}$.

Exercice 5. *Coloriages*

Un graphe est k-coloriable si on peut colorier ses sommets en utilisant au maximum k couleurs, de sorte que deux sommets adjacents aient des couleurs différentes.

1. Montrer que le graphe suivant est 3-coloriable mais pas 2-coloriable :



☞ G n'est pas 2-coloriable parce qu'il contient des 3-cliques (des triangles). Par contre, il est 3-colorié par la fonction suivante : $\begin{array}{c|c|c|c|c|c} A & B & C & D & E & F \\ \hline 1 & 2 & 3 & 2 & 3 & 1 \end{array}$.

2. Montrer que le problème suivant est dans NP :

   3-COLOR
   Entrée : Un graphe G ;
   Question : G est-il 3-coloriable ?

☞ 3-COLOR admet l'algorithme de vérification polynomial suivant : Sur l'entrée $G = (V, E)$:

(i) Deviner une fonction $c : V \to \{1,2,3\}$;

(ii) Vérifier que c constitue un coloriage de G.

Alors :
- L'objet c deviné est bien de taille polynomialement bornée en celle de G, puisqu'il peut être vu comme un n-uplet à valeurs dans $\{1,2,3\}$ (d'où $\mathsf{taille}(c) = |V| \leq \mathsf{taille}(G)$).
- L'étape (ii) se fait en temps $O(n+m)$: il suffit de parcourir le graphe et de tester l'inégalité $c(x) \neq c(y)$ pour chaque arête xy.

Exercice 6. DOUBLE-SAT

On considère le problème suivant :

DOUBLE-SAT
Entrée : Une formule propositionnelle φ ;
Question : φ admet-elle (au moins) deux modèles ?

1. Montrer que DOUBLE-SAT $\in$ NP.
☞ DOUBLE-SAT est décidé par la procédure non-déterministe suivante :

(i) Deviner un couple d'évaluations (T, T') sur les variables de φ ;

(ii) vérifier que ces deux évaluations sont distinctes et satisfont φ.

Or :

- l'objet (T, T') deviné est de taille polynômialement (et même linéairement) bornée par celle de φ (chaque évaluation est un sous ensemble de l'ensemble $\mathsf{var}(\varphi)$ et donc est de taille inférieure à l'entier $|\mathsf{var}(\varphi)|$ qui est lui même inférieur à $\mathsf{taille}(\varphi)$ (nombre d'occurences de variables dans φ). Donc $\mathsf{taille}(T, T') \leq 2\mathsf{taille}(\varphi)$.
- les vérifications de la phase (ii) se font en un temps (déterministe) polynômial en la taille de φ (vu en TD).

Finalement, cette procédure est non-déterministe polynômiale et on a bien DOUBLE-SAT $\in$ NP.

2. Montrer que SAT se réduit polynômialement à DOUBLE-SAT.
☞ À chaque $\varphi \in \mathscr{I}(\text{SAT})$ on associe une formule $r(\varphi)$ de la manière suivante :

- on choisit une variable x n'apparaissant pas dans φ ;
- on pose $r(\varphi) = \varphi \wedge (x \vee \neg x)$.

Alors $r(\varphi) \in \mathscr{I}(\text{DOUBLE-SAT})$ (*i.e.* $r(\varphi)$ est une formule propositionnelle) et on a : φ est satisfaisable ssi $r(\varphi)$ a deux modèles distincts. En effet, si T est un modèle de φ alors les extensions de T à $\mathsf{var}(\varphi) \cup \{x\}$ obtenues en affectant, respectivement, x ou $\neg x$ à VRAI sont deux modèles distincts de $r(\varphi)$. Inversement, si $r(\varphi)$ a deux modèles, alors la restriction de l'un quelconque d'entre eux à $\mathsf{var}(\varphi)$ est un modèle de φ. Par conséquent, r est une application de $\mathscr{I}(\text{SAT})$ dans $\mathscr{I}(\text{DOUBLE-SAT})$ qui préserve la positivité des instances : c'est une réduction. De plus, $\mathsf{taille}(r(\varphi))$ est clairement polynômialement bornée par $\mathsf{taille}(\varphi)$ et l'application r est trivialement calculable en temps déterministe polynômial. Finalement, r est une réduction polynômiale et on a bien : SAT $\leq_{\text{pol}}$ DOUBLE-SAT.

3. Conclusion ?
☞ Conclusion : DOUBLE-SAT $\in$ NP, SAT $\leq_{\text{pol}}$ DOUBLE-SAT et SAT est NP-complet : DOUBLE-SAT est NP-complet (voir Théorème 21theorem.21 du Cours).