Examen du cours Complexité (L3)

Les documents (notes, polycopiés, ..) et ordinateurs (calculatrices, téléphones, tablettes, ..) ne sont pas autorisés.

Date : 10 janv. 2022 à 10h45 / Durée : 2 heures

Les questions sont en général simples et on peut y répondre correctement et rigoureusement en quelque lignes. Seules les questions marquées d'une étoile demandent une construction un peu plus longue à décrire et/ou une preuve un peu plus difficile. En conséquence, ces questions apportent plus de points.

On considère des variantes du problème de savoir si une formule propositionnelle en forme normale conjonctive (CNF) est satisfaisable.

Rappelons qu'une formule CNF est de la forme $\phi \equiv C_1 \wedge C_2 \wedge \cdots \wedge C_m$ où chaque clause C_i est de la forme $C_i \equiv l_{i,1} \vee l_{i,2} \vee \cdots \vee l_{i,r_i}$ où chaque littéral $l_{i,j}$ est soit une variable x_k soit sa négation $\neg x_k$. Notons que $m = 0$ (pas de clauses) ou $r_i = 0$ (la clause C_i est vide) sont autorisés.

On s'autorise à écrire $\neg l$ pour un littéral l, avec la convention que $\neg\neg x$ désigne x, de sorte que la négation d'un littéral est encore un littéral. La taille d'une clause, notée $|C|$, est le nombre de littéraux *distincts* dans C (car les clauses sont souvent représentées par des ensembles), de sorte que l'on ne distingue pas entre, par exemple, $x \vee \neg x$ et $x \vee \neg x \vee x$. Une formule CNF est une k-CNF si chacune de ses clauses est de taille au plus k. Elle est strict-k-CNF si chaque clause est de taille exactement k. On note $Var(\phi) = \{x_1, \ldots, x_n\}$ pour l'ensemble des variables propositionnelles qui apparaissent dans ϕ.

$\mathsf{2SAT}$ est le problème de savoir si une 2-CNF ϕ est satisfaisable. Formellement, $\mathsf{2SAT}$ est donc le langage des 2-CNF satisfaisables (écrites dans une notation et avec un alphabet qu'on laisse non spécifiés). De même $\mathsf{strict2SAT}$ est le problème de savoir si une strict-2-CNF est satisfaisable.

Partie 1 : de coGAP à 2SAT.

1. Définissez une réduction logspace montrant que $\mathsf{2SAT} \leq \mathsf{strict2SAT}$. Énoncez et démontrez sa correction.

Solution:

Pour un input x, notre réduction r vérifie d'abord que x est bien une formule $\mathsf{2SAT}$ et sinon renvoie x inchangé. Si $x = \phi = \bigwedge_{i=1}^{m} C_i$ avec $Var(\phi) = \{x_1, \ldots, x_n\}$, on construit $r(x) = \phi'$ obtenu à partir de ϕ en (1) conservant les clauses de taille 2, (2) remplaçant les clauses $l_{i,1}$ de taille 1 par les deux clauses $(l_{i,1} \vee y) \wedge (l_{i,1} \vee \neg y)$ qui utilisent une nouvelle variable y, (3) remplaçant les clauses de taille 0 (qui sont insatisfaisables) par les quatre clauses $(y \vee y') \wedge (y \vee \neg y') \wedge (\neg y \vee y') \wedge (\neg y \vee \neg y')$ avec y' une seconde nouvelle variable. (NB : On peut aussi, dans le cas d'une clause vide, renvoyer comme valeur pour $r(x)$ un mot quelconque qui ne soit pas une CNF. C'est plus efficace mais respecte moins l'esprit des réductions).

On vérifie facilement que r est logspace. La correction s'énonce $x \in \mathsf{2SAT}$ ssi $r(x) \in \mathsf{strict2SAT}$ et elle est facile à démontrer (démonstration omise ici).

Soit un graphe orienté $G = (V, E)$. Pour deux sommets $s, s' \in V$ on note $s \xrightarrow{*}_G s'$, et on dit que s' est atteignable depuis s dans G, s'il existe un chemin $s \to \cdots \to s'$ utilisant les arêtes de G. On écrit $s \xrightarrow{*} s'$ sans l'indice G quand le contexte est clair. Notons qu'on a $s \xrightarrow{*} s$ (via un chemin de longueur 0) pour tout sommet $s \in V$.

À $G = (V, E)$ on associe un ensemble de variables $V_G = \{x_s \mid s \in V\}$, c.-à-d. une variable par sommet. À chaque arête $e \in E$ de G on associe une clause C_e : si e est $s \to s'$ alors on pose $C_e = \neg x_s \vee x_{s'}$. On pose enfin $\phi_G = \bigwedge_{e \in E} C_e$.

2. Soit $v : V_G \to \{\top, \bot\}$ une valuation qui valide ϕ_G. Montrez que si $v(x_s) = \top$ et si s' est atteignable depuis s, alors $v(x_{s'}) = \top$.

Solution:
On prouve la propriété pour des sommets quelconques s, s' par induction sur la longueur d'un chemin $s \xrightarrow{*} s'$. Pour un chemin de longueur 0, on a $s' = s$. Un chemin de longueur > 0 a la forme $s \to s_1 \xrightarrow{*} s'$ où $s \to s_1$ est une arête e de G. Alors ϕ_G contient $C_e = \neg x_s \vee x_{s_1}$ et puisque $v(C_e) = \top$ et $v(x_s) = \top$ on a nécessairement $v(x_{s_1}) = \top$. En raisonnant maintenant sur le chemin $s_1 \xrightarrow{*} s'$, l'hypothèse d'induction nous donne $v(x_{s'}) = \top$.

Pour $W \subseteq V$ un sous-ensemble de sommets, on définit v_W comme étant la valuation telle que $v_W(x_s) = \top$ ssi s est atteignable depuis un sommet de W.

3. Montrez que pour tout $W \subseteq V$, v_W valide ϕ_G.

Solution:
Soit une clause $C_e \equiv \neg x_s \vee x_{s'}$ de ϕ_G, correspondant à l'arête $s \to s'$. Si s est atteignable depuis W alors s' l'est aussi donc $v_W(x_{s'}) = \top$. Si s n'est pas atteignable depuis W alors $v_W(x_s) = \bot$, donc $v_W(\neg x_s) = \top$. Dans les deux cas v_W valide C_e. Puisque ce raisonnement vaut pour chaque clause de ϕ_G, v_W valide ϕ_G.

4. Montrez que coGAP $\leq$ strict2SAT, où coGAP est le complémentaire du *Graph Accessibility Problem* vu en cours, c.-à-d. que coGAP est l'ensemble des (G, s, t) tels que $s \overset{*}{\not\to}_G t$.

Solution:
On commence par réduire coGAP à 2SAT. Notre réduction associe à (G, s, t) la formule ψ définie par $\psi \equiv \phi_G \wedge x_s \wedge \neg x_t$. Il s'agit bien d'une conjonction de clauses ayant au plus deux littéraux, et on peut la construire en logspace. La correction s'énonce :
Lemme. ψ est satisfaisable ssi G n'a pas de chemin de s à t.
Preuve. ($\Longleftarrow$) : Supposons que t n'est pas atteignable depuis s et soit W l'ensemble des sommets atteignables depuis s. On a vu (Question 3) que v_W satisfait ϕ_G. Par ailleurs $v_W(x_s) = \top$ et $v_W(x_t) = \bot$. Donc v_w valide aussi les clauses x_s et $\neg x_t$, donc il valide ψ, donc ψ est satisfaisable.
($\Longrightarrow$) : Si $v(\psi) = \top$ alors $v(x_s) = \top$. On a vu (Question 2) qu'alors $v(x_{s'}) = \top$ pour tout s' atteignable depuis s. Or $v(\psi) = \top$ exige $v(x_t) = \bot$. Donc t n'est pas atteignable. $\square$
Finalement, on a montré coGAP $\leq$ 2SAT, et avec la Question 1 on obtient coGAP $\leq$ strict2SAT.

Partie 2 : de 2SAT à GAP

À une strict-2-CNF ϕ on associe le graphe orienté $G_\phi = (V_\phi, E_\phi)$ dont les sommets sont les $2n$ littéraux associés aux n variables $x_1, \ldots, x_n$ de $Var(\phi)$ et dont les arêtes sont toutes les paires $(\neg l) \to l'$ et $(\neg l') \to l$ telles que ϕ contient une clause $l \vee l'$.

5. Soient l, l' deux littéraux quelconques. Montrez que si $l \xrightarrow{*} l'$ dans G_ϕ alors on a aussi $\neg l' \xrightarrow{*} \neg l$.

Solution:

La propriété est vraie pour les chemins de longueur 1, i.e., les arêtes : si $l_0 \to l_1$ est une arête alors, par définition de G_ϕ, $\neg l_0 \vee l_1$ ou $l_1 \vee \neg l_0$ est une clause de ϕ, et dans les deux cas on déduit, toujours par définition de G_ϕ, que $\neg l_1 \to \neg l_0$ est une arête. La propriété est aussi vraie pour les chemins de longueur 0 : forcément $l = l'$, donc $\neg l' = \neg l$, donc $\neg l' \xrightarrow{*} \neg l$. Si maintenant on a $l \xrightarrow{*} l'$ via les arêtes $l \to l_1$, $l_1 \to l_2$, ..., $l_n \to l'$ on a aussi les arêtes $\neg l' \to \neg l_n$, ..., $\neg l_2 \to \neg l_1$, $\neg l_1 \to \neg l$, soit $\neg l' \xrightarrow{*} \neg l$.

6. Montrez que si ϕ est satisfaisable, et si $l_0 \to l_1 \to \cdots \to l_n = l_0$ est un circuit dans G_ϕ alors $l_i \neq \neg l_j$ pour tous $0 \leq i, j \leq n$, c.-à-d. que le circuit ne contient pas deux littéraux opposés.

Solution:

Puisqu'il existe des valuations qui satisfont ϕ, nous en fixons une, disons v.
Lemme. Si $v(l) = \top$ et si $l \to l'$ est une arête, alors $v(l') = \top$.
Preuve. Puisque $l \to l'$ est une arête, ϕ contient la clause $C = \neg l \vee l'$ (ou son symétrique $l' \vee \neg l$). Puisque $v(\phi) = \top$ alors en particulier $v(C) = \top$, soit $v(\neg l \vee l') = \top$. Mais puisque $v(l) = \top$ par hypothèse, c.-à-d. $v(\neg l) = \bot$, il faut nécessairement $v(l') = \top$. $\Box$

Maintenant supposons (raisonnement par l'absurde) qu'il existe un chemin $l_0 \to \cdots \to l_n$ avec $l_i = \neg l_j$. Puisqu'ils sont opposés, l'un parmi l_i et l_j est validé par v (et l'autre est invalidé). Puisque le circuit contient un littéral validé par v, le Lemme nous dit alors que tous les littéraux du circuit sont validés par v, contredisant l'hypothèse $l_i = \neg l_j$.

(*) 7. Montrez que si G_ϕ ne contient pas de circuit où figurent un littéral et son opposé, alors ϕ est satisfaisable. Idée : on pourra construire étape par étape une valuation à partir du graphe. Pour cette construction, il faut donner les propriétés vérifiées à chaque étape et montrer qu'elles restent préservées.

Solution:

Partons d'une valuation $v_0 = []$ dont le domaine de définition est vide et étendons la par étapes $v_0 \curvearrowright v_1 \curvearrowright v_2 \curvearrowright \cdots \curvearrowright v_k$ jusqu'à ce que le domaine de v_k contienne toutes les variables de ϕ. À chaque étape, v_i respecte la propriété suivante (« l'invariant ») :

saturation : Soit $l \to l'$ une arête de G_ϕ : si $v_i(l) = \top$ alors $v_i(l') = \top$.

Notons que cet invariant est trivialement respecté par v_0 puisque $dom(v_0) = \emptyset$, donc $v_0(l)$ n'est jamais défini.

Plaçons nous à l'étape i et supposons qu'il existe une variable x qui n'appartient pas encore à $dom(v_i)$. On fixe une telle variable et on construit v_{i+1} en étendant v_i.

1er cas : Si G_ϕ a un chemin $x \to \cdots \to \neg x$, on pose $v_{i+1}(x) = \bot$, donc $v_{i+1}(\neg x) = \top$, et on pose plus généralement $v_{i+1}(l) = \top$ pour tout littéral l tel qu'il existe un chemin $\neg x \to \cdots \to l$, ce qui garantit la propriété de saturation. Vérifions que cette extension n'est pas en conflit avec les valeurs déjà définies dans v_i. La définition $v_{i+1}(x) = \bot$ ne crée pas de conflit car $x \notin dom(v_i)$. Considérons ensuite les nouvelles définitions « $v_{i+1}(l) = \top$ » introduites par la saturation. Tout d'abord on ne peut pas avoir $l = x$ car s'il y avait un chemin de $\neg x$ à $l = x$, on aurait un circuit avec x et sa négation. Ensuite si $v_i(l)$ était déjà défini alors forcément on avait $v_i(l) = \top$ (évitant le conflit avec $v_{i+1}(l)$). En effet, si on avait $v_i(l) = \bot$, alors $v_i(\neg l) = \top$. Comme $\neg l \to \cdots \to x$ (cf. Question 2), on avait $v_i(x) = \top$ par l'hypothèse de saturation : contradiction. Finalement v_{i+1} est bien une extension de v_i. Par construction elle vérifie l'hypothèse de saturation.

2e cas : Si G_ϕ a un chemin $\neg x \to \cdots \to x$, on fait la construction symétrique : on pose $v_{i+1}(x) = \top$ et on sature en posant $v_{i+1}(l) = \top$ pour tous les l tels que $x \to \cdots \to l$.

3e cas : Si G_ϕ n'a ni chemin $\neg x \to \cdots \to x$ ni chemin $x \to \cdots \to \neg x$ on fait (au choix) comme dans le 1er ou le 2e cas. L'étape de saturation ne crée pas de conflit et v_{i+1} est une extension de v_i.

Il ne reste plus qu'à montrer que, quand on arrive à une valuation v_k tel que $dom(v_k)$ contienne toutes les variables de ϕ, alors $v(\phi) = \top$. Prenons une 2-clause $C \equiv l \vee l'$. Si $v(l) = \top$, C est validée par v. Si $v(l) = \bot$ alors $v(\neg l) = \top$. Puisque G_ϕ a une arête $\neg l \to l'$, on a $v(l') = \top$ par hypothèse de saturation. Donc $v'(C) = \top$ ici aussi. Ce raisonnement vaut pour toutes les clauses, donc $v(\phi) = \top$, donc ϕ est satisfaisable.

8. Montrez qu'on peut décider strict-2-CNF en temps polynomial. (Indication : n'inventez pas un algorithme nouveau dont il faudrait prouver la correction, ramenez vous plutôt à un problème d'algorithmique de graphes sur G_ϕ.)

Solution:
Étant donné une strict-2-CNF ϕ, on construit G_ϕ et on vérifie pour chaque variable x que soit $\neg x$ n'est pas atteignable depuis x, soit x n'est pas atteignable depuis $\neg x$. La correction de cet algorithme est donnée par les deux questions précédentes. On a donc à résoudre $2n$ questions d'accessibilité sur un graphe construit en temps polynomial : c'est faisable en temps polynomial.

(*) 9. Adaptez la construction plus haut « $\phi \mapsto G_\phi$ » pour obtenir une réduction de strict2SAT à coGAP.

Solution:
Soit ϕ une strict-2-CNF sur n variables et G_ϕ le graphe à $2n$ sommets associé.

On construit un graphe H_ϕ composé de $2n$ copies de G_ϕ (une par littéral) qu'on nommera donc $G^{x_1}, G^{\neg x_1}, G^{x_2}, \ldots, G^{x_n}, G^{\neg x_n}$, ainsi que de deux sommets supplémentaires s et t. H a donc $4n^2 + 2$ sommets. On relie ces composants avec :

- pour chaque variable x de ϕ, une arête $s \to s_x^x$ reliant s au sommet s_x de la copie G^x ;
- pour chaque variable x, une arête $s_{\neg x}^x \to s_{\neg x}^{\neg x}$ permettant de passer de G^x à $G^{\neg x}$;
- pour chaque variable x, une arête $s_x^{\neg x} \to t$ reliant le sommet s_x de $G^{\neg x}$ à t.

Preuve de correction : On ne peut aller de s à t dans H_ϕ qu'en choisissant une variable x, en allant de s à s_x^x dans G^x, puis en traversant G^x de s_x^x à $s_{\neg x}^x$, permettant de passer dans $G^{\neg x}$ via $s_{\neg x}^x \to s_{\neg x}^{\neg x}$, puis en traversant $G^{\neg x}$ de $s_{\neg x}^{\neg x}$ à $s_x^{\neg x}$, puis en rejoignant t.

Donc H_ϕ a un chemin de s à t ssi dans G_ϕ on peut aller de s_x à $s_{\neg x}$ et de $s_{\neg x}$ à s_x pour une certaine variable x, c.-à-d. ssi ϕ n'est pas satisfaisable (cf. Question 8). Par contraposition, ϕ est satisfaisable ssi H_ϕ n'a pas de chemin $s \xrightarrow{*} t$. La réduction $\phi \mapsto (H_\phi, s, t)$ réduit donc strict2SAT à coGAP. On vérifie facilement qu'elle est logspace.

10. Montrez que 2SAT est NL-complet.

Solution:
On a montré coGAP $\leq$ strict2SAT $\leq$ coGAP. On sait aussi que strict2SAT $\leq$ 2SAT $\leq$ strict2SAT. Donc 2SAT, strict2SAT et coGAP sont équivalents. On conclut puisque coGAP est coNL-complet donc NL-complet (théorème d'Immerman-Szelepscényi).

Partie 3 : de 3SAT à MAJ2SAT.

Le langage MAJ2SAT est constitué des formules 2-CNF qui sont *majoritairement satisfaisables*, c.-à-d., telles qu'il existe une valuation satisfaisant *au moins les deux tiers* de leurs clauses (il s'agit donc d'une « majorité qualifiée »).

11. Donnez une réduction prouvant 2SAT $\leq$ MAJ2SAT.

Solution:
Soit une variable nouvelle y et ψ_y la conjonction $y \wedge \neg y$ de deux clauses. Une valuation quelconque satisfait toujours exactement une clause de ψ_y. Donc si v satisfait exactement k parmi les m clauses de ϕ, v satisfait $k+1$ clauses parmi les $m+2$ clauses de $\phi \wedge \psi_y$ et $k+\ell$ clauses parmi les $m+2\ell$ clauses de $\phi \wedge \psi_{y_1} \wedge \cdots \wedge \psi_{y_\ell}$, une formule qu'on va appeler ϕ'_ℓ. Finalement, v satisfait *toutes* les clauses de ϕ ssi elle satisfait $m+\ell$ clauses de ϕ'_ℓ . En prenant $\ell = m$, on a bien que ϕ est satisfaisable ssi au moins deux tiers des clauses de ϕ'_m sont simultanément satisfaisables ssi ϕ'_m est dans $\mathsf{MAJ2SAT}$.

(*) 12. Donnez une réduction prouvant $\mathsf{3SAT} \leq \mathsf{MAJ2SAT}$.

Solution:
Puisque $\mathsf{3SAT} \leq \mathsf{strict3SAT}$ il suffit de prouver $\mathsf{strict3SAT} \leq \mathsf{MAJ2SAT}$ ce qui simplifie les notations. Une instance de $\mathsf{strict3SAT}$ est de la forme $\phi = \bigwedge_{i=1}^m C_i$ où chaque C_i est une conjonction de 3 littéraux $C_i = a_i \vee b_i \vee c_i$ distincts. On associe à chaque C_i une conjonction D_i de 10 clauses, définie par :

$$D_i \equiv E_i \wedge F_i \wedge G_i \equiv \begin{cases} & \overbrace{a_i \wedge b_i \wedge c_i \wedge y_i}^{E_i \text{ (4 clauses)}} \\ \wedge & \overbrace{(\neg a_i \vee \neg b_i) \wedge (\neg a_i \vee \neg c_i) \wedge (\neg b_i \vee \neg c_i)}^{F_i \text{ (3 clauses)}} \\ \wedge & \overbrace{(a_i \vee \neg y_i) \wedge (b_i \vee \neg y_i) \wedge (c_i \vee \neg y_i)}^{G_i \text{ (3 clauses)}} \end{cases}$$

où, pour chaque i, y_i est une nouvelle variable. Il s'agit bien de 2-clauses.

Lemme. Une valuation v quelconque satisfait au plus 7 clauses de D_i et toute valuation qui satisfait C_i peut être étendue (sur y_i) de façon à satisfaire 7 clauses de D_i. Une valuation qui ne satisfait pas C_i satisfait au plus (suivant la valeur de $v(y_i)$) 6 clauses de D_i.

Preuve (idée). On examine tous les cas possibles en tirant parti des symétries de D_i :
- Si v valide les 3 littéraux a_i, b_i, et c_i, alors E_i et G_i sont validées mais aucune clause de F_i n'est satisfaite.
- Si v valide exactement 2 littéraux parmi a_i, b_i et c_i, alors il valide aussi 2 clauses de E_i, 2 de F_i et 2 de G_i plus une septième dans E_i ou G_i suivant la valeur de $v(y_i)$.
- etc. $\square$

On définit alors $\phi' \equiv D_1 \wedge \cdots \wedge D_m \wedge \psi_{z_1} \wedge \cdots \wedge \psi_{z_m}$ (avec $\psi_z \equiv z \wedge \neg z$ comme à la question précédente) et on obtient une conjonction de $12m$ 2-clauses. Pour en valider au moins $8m$, il faut valider au moins $7m$ clauses parmi $D_1 \wedge \cdots \wedge D_m$, donc il faut une valuation qui valide exactement 7 clauses de chaque D_i (on ne peut pas faire mieux).

13. Montrez que $\mathsf{MAJ2SAT}$ est NP-complet.

Solution:
La question 12 a montré que $\mathsf{MAJ2SAT}$ est NP-difficile. Il ne reste plus qu'à vérifier que le problème est dans NP, ce qui est évident : il suffit de deviner une valuation et on vérifie ensuite en temps polynomial.