Examen Complexité 2013/2014

On portera un soin particulier à montrer lors des réductions que les problèmes considérés sont équivalents.

Partie 1 : 2SAT est NLOGSPACE-complet.

On considère une formule $\varphi = \wedge_{j=1}^{m} a_j \vee b_j$ où les a_j et les b_j sont des littéraux construits à partir des propositions atomiques $p_1, \ldots, p_n$. Une clause $a_j \vee b_j$ est équivalente à $\neg a_j \Rightarrow b_j$ et à $\neg b_j \Rightarrow a_j$. Soit G_φ un graphe orienté dont les sommets sont les littéraux et les arcs sont $\{(\neg a_j, b_j), (\neg b_j, a_j)\}_{j \leq m}$.

Question 1. Démontrer que φ n'est pas satisfaisable si et seulement si, dans G_φ, il existe un circuit contenant une proposition atomique et sa négation.

Question 2. Démontrer que le problème 2SAT est dans NLOGSPACE.

On considère un graphe orienté G et deux sommets s et t tels que s n'a pas de prédécesseur et t n'a pas de successeur.

Question 3. Démontrer que le problème de l'accessibilité de t depuis s est réductible en espace logarithmique à un problème de 2SAT.
Indication : les propositions atomiques de la formule sont les sommets du graphe.

Partie 2 : 3-Dimensional Matching est NP-complet.

Le 3-Dimensional Matching est défini par :
- trois ensembles finis disjoints de même taille L, A et B ;
- un sous-ensemble de triplets $M \subseteq L \times A \times B$.

Le problème consiste à décider s'il existe $M' \subseteq M$ tel que tout élément de $L \cup A \cup B$ apparaît dans exactement un triplet de M'. Un tel M' s'appelle un matching.

Question 4. Montrer que le 3-Dimensional Matching est dans NP.

On considère une formule $\varphi = \wedge_{j=0}^{m-1} l_{j,1} \vee l_{j,2} \vee l_{j,3}$ où les $l_{j,k}$ sont des littéraux construits à partir des propositions atomiques $p_1, \ldots, p_n$.

On associe à φ le 3-Dimensional Matching problème suivant.
- $L = \{p_{i,j}, \overline{p}_{i,j}\}_{1 \leq i \leq n, 0 \leq j < m}$,
  $A = \{a_{i,j}\}_{1 \leq i \leq n, 0 \leq j < m} \cup \{a'_j\}_{0 \leq j < m} \cup \{a''_k\}_{1 \leq k \leq m(n-1)}$
  et $B = \{b_{i,j}\}_{1 \leq i \leq n, 0 \leq j < m} \cup \{b'_j\}_{0 \leq j < m} \cup \{b''_k\}_{1 \leq k \leq m(n-1)}$;
- $M = \mathit{Set} \cup \mathit{Check} \cup \mathit{Fill}$ avec :
  1. $\mathit{Set} = \{(p_{i,j}, a_{i,j}, b_{i,j}), (\overline{p}_{i,j}, a_{i,j\oplus 1}, b_{i,j})\}_{1 \leq i \leq n, 0 \leq j < m}$
     où $\oplus$ désigne l'addition modulo m.
  2. $\mathit{Check} = \{(l_{j,s}, a'_j, b'_j)\}_{0 \leq j < m, 1 \leq s \leq 3}$
  3. $\mathit{Fill} = \{(p_{i,j}, a''_k, b''_k), (\overline{p}_{i,j}, a''_k, b''_k)\}_{1 \leq i \leq n, 0 \leq j < m, 1 \leq k \leq m(n-1)}$

Question 5. Montrer que ce problème noté P_φ peut être construit en temps polynomial.

Question 6. Montrer que si φ est satisfaisable alors P_φ admet un matching.

Question 7. Soit M' un matching de P_φ. Montrer que pour tout i, j :
- exactement l'un des deux triplets $(p_{i,j}, a_{i,j}, b_{i,j})$, $(\overline{p}_{i,j}, a_{i,j\oplus 1}, b_{i,j})$ doit appartenir à M' ;
- si un triplet $(p_{i,j}, a_{i,j}, b_{i,j}) \in M'$ (resp. $(\overline{p}_{i,j}, a_{i,j\oplus 1}, b_{i,j}) \in M'$) alors pour tout j', $(p_{i,j'}, a_{i,j'}, b_{i,j'}) \in M'$ (resp. $(\overline{p}_{i,j'}, a_{i,j'\oplus 1}, b_{i,j'}) \in M'$) ;

– exactement l'un des trois triplets $(l_{j,1}, a'_j, b'_j), (l_{j,2}, a'_j, b'_j), (l_{j,3}, a'_j, b'_j)$ doit appartenir à M'.

Question 8. En déduire que si P_φ admet un matching alors φ est satisfaisable.

Partie 3 : La vacuité des expressions rationnelles étendues est PSPACE-complet.

Soit Σ un alphabet, on considère des expressions rationnelles *étendues* c'est à dire composées du mot vide ε, des lettres de Σ, des opérateurs de concaténation $\cdot$, d'union $+$, d'itération * et d'intersection $\cap$.
Par exemple avec $\Sigma = \{a, b\}$, l'expression $(b^* \cdot a \cdot (a+b)^*) \cap (a^* \cdot b \cdot (a+b)^*)$ désigne l'ensemble des mots qui contiennent au moins une occurrence de a et de b.

Etant donnée une expression E (resp. un automate $\mathcal{A}$), on notera $L(E)$ (resp. $L(\mathcal{A})$) le langage associé à E (resp. $\mathcal{A}$).

On rappelle qu'un problème de planification est donné par n variables booléennes $\{x_i\}_{i \in I}$ avec $I = \{1, \ldots, n\}$, m opérations où chaque opération est définie par une condition de la forme $\bigwedge_{i \in I'} x_i = \alpha_i$ avec $I' \subseteq I$ et une mise à jour de la forme $\{x_i \leftarrow \beta_i\}_{i \in I''}$ avec $I'' \subseteq I$, une configuration initiale s_{init} et une configuration finale s_{fin}.
Une opération est applicable à une configuration si la condition de l'opération est satisfaite. Son application consiste à effectuer ses mises à jour pour obtenir la nouvelle configuration.
Le problème consiste à déterminer s'il existe une suite d'applications des opérations qui conduise de la configuration initiale à la configuration finale.

Question 9. En vous servant de l'exemple ci-dessous montrer qu'un problème de planification se réduit en temps polynomial à la vacuité du langage d'une expression régulière étendue. L'expression E produite aura la forme $E = E_1 \cap \ldots \cap E_n$ où E_i décrit les suites d'applications qui conduisent au succès *en ne considérant que la variable* x_i.

$s_{init} : x_1 = x_2 = x_3 = \bot$	
$s_{fin} : x_1 = x_2 = x_3 = \top$	$((o_2^* \cdot o_3 \cdot o_1^* \cdot o_2)^* \cdot o_2^* \cdot o_3 \cdot o_1^*)$
	$\cap$
o_1 : **If** x_1 **and not** x_2 **then** $x_3 \leftarrow \top$	$((o_1^* \cdot o_3 \cdot o_3^* \cdot o_2)^* \cdot o_1^* \cdot o_3 \cdot o_3^*)$
o_2 : **If** x_2 **and** x_3 **then** $x_1 \leftarrow \bot; x_2 \leftarrow \bot$	$\cap$
o_3 : **If not** x_1 **then** $x_1 \leftarrow \top; x_2 \leftarrow \top$	$(o_3^* \cdot o_1 \cdot (o_1 + o_2 + o_3)^*)$

Question 10. Soient deux expressions rationnelles E_1 et E_2 et deux automates non déterministes $\mathcal{A}_1$ et $\mathcal{A}_2$ tels que pour $i \in \{1, 2\}$, $L(E_i) = L(\mathcal{A}_i)$. Montrer que :

– on peut construire un automate $\mathcal{A}$ tel que $L(E_1 \cdot E_2) = L(\mathcal{A})$ avec $|S| = |S_1| + |S_2|$
– on peut construire un automate $\mathcal{A}$ tel que $L(E_1 + E_2) = L(\mathcal{A})$ avec $|S| = |S_1| + |S_2|$
– on peut construire un automate $\mathcal{A}$ tel que $L(E_1 \cap E_2) = L(\mathcal{A})$ avec $|S| = |S_1||S_2|$
– on peut construire un automate $\mathcal{A}$ tel que $L(E_1^*) = L(\mathcal{A})$ avec $|S| \leq |S_1| + 1$

Question 11. En déduire que si $L(E)$ est non vide alors $L(E)$ contient un mot de longueur au plus $2^{|E|}$.

On observe qu'étant donné un mot w :

- $w \in L(E_1 \cap E'_1)$ et ... et $w \in L(E_k)$ ssi
  $w \in L(E_1)$ et $w \in L(E'_1)$ et ... et $w \in L(E_k)$
- $w \in L(E_1 + E'_1)$ et ... et $w \in L(E_k)$ ssi
  $w \in L(E_1)$ et ... et $w \in L(E_k)$ ou $w \in L(E'_1)$ et ... et $w \in L(E_k)$
- $w \in L(E_1^*)$ et ... et $w \in L(E_k)$ ssi
  $w \in L(E_1 \cdot E_1^*)$ et ... et $w \in L(E_k)$ ou $w = \varepsilon$ et ... et $\varepsilon \in L(E_k)$
- $w \in L((E_1 \cap E'_1) \cdot E''_1)$ et ... et $w \in L(E_k)$ ssi
  $w \in L(E_1 \cdot E''_1)$ et $w \in L(E'_1 \cdot E''_1)$ et ... et $w \in L(E_k)$ ou
  $\varepsilon \in L(E_1)$ et $\varepsilon \in L(E'_1)$ et $w \in L(E''_1)$ et ... et $w \in L(E_k)$
- $w \in L((E_1 + E'_1) \cdot E''_1)$ et ... et $w \in L(E_k)$ ssi
  $w \in L(E_1 \cdot E''_1)$ et ... et $w \in L(E_k)$ ou
  $w \in L(E'_1 \cdot E''_1)$ et ... et $w \in L(E_k)$ ou
  $\varepsilon \in L(E_1)$ et $w \in L(E''_1)$ et ... et $w \in L(E_k)$ ou
  $\varepsilon \in L(E'_1)$ et $w \in L(E''_1)$ et ... et $w \in L(E_k)$
- $w \in L(E_1^* \cdot E'_1)$ et ... et $w \in L(E_k)$ ssi
  $w \in L(E_1 \cdot E_1^* \cdot E'_1)$ et ... et $w \in L(E_k)$ ou
  $w \in L(E'_1)$ et ... et $w \in L(E_k)$
- $w \in L(a \cdot E_1)$ et ... et $w \in L(a \cdot E_k)$ ssi
  $w = aw'$ et $w' \in L(E_1)$ et ... et $w' \in L(E_k)$

Question 12. En utilisant les observations précédentes, proposer une procédure *non déterministe* en espace polynomial par rapport à $|E|$ pour tester si un mot $w \in L(E)$. Cette procédure maintiendra une conjonction de contraintes $\bigwedge_{1 \leq j \leq m_1} \varepsilon \in E_j \wedge \bigwedge_{m_1+1 \leq j \leq m_1+m_2} w' \in E_j$ avec w' un suffixe de w. On montrera en particulier :

- comment traiter le cas $\varepsilon \in E_j$.
- que le nombre de contraintes à tester $m_1 + m_2$ est au plus égal à $2(n_\cap^E + 1)$ où $n_\cap^E$ est le nombre d'occurrences d'intersection de E.
- pour tout j, que la longueur de l'expression E_j est au plus égale à $(n_*^E + 1)|E|$ où n_*^E est le nombre d'occurrences d'itération de E.
  (la preuve de ce point est difficile)

Question 13. Montrer que le problème de la vacuité du langage $L(E)$ où E est une expression rationnelle étendue est dans PSPACE.
Indication : On pourra utiliser les questions 11 et 12 pour proposer une procédure non déterministe en précisant à quel moment on devine les lettres du mot qui pourrait appartenir à $L(E)$.