Complexité - compléments

`Exercice 1` `Isomorphisme de graphes`

Soit deux graphes $G_1 = (V, E_1)$ et $G_2 = (V, E_2)$ qui ont le même ensemble de sommets $V = \{1, 2, \ldots, n\}$. On dit que les deux graphes sont isomorphes s'il existe une permutation $\sigma \in S_n$ telle que $(i, j) \in E_1$ ssi $(\sigma(i), \sigma(j)) \in E_2$. Deux graphes ne sont pas isomorphes s'il n'existe pas d'isomorphisme d'un graphe vers l'autre.

1. Montrer que le langage $GI = \{\langle G, H\rangle \mid G \text{ et } H \text{ sont isomorphes}\}$ est dans NP.

2. Que pouvez-vous dire pour le complémentaire de ce langage, appelé GNI ?

On dit qu'un système de preuve est un protocole (échange de messages) entre deux joueurs, un vérifieur V et un prouveur P qui sont des machines de Turing.

- Le vérifieur est en plus randomisé et fonctionne en temps polynomial. (Il a accès à une bande qui contient des bits aléatoires.)
- Le prouveur est tout puissant mais n'a pas accès aux bits aléatoires du vérifieur, seulement ceux révélés au cours des échanges.

À la fin du protocole, le vérifieur doit être convaincu d'un théorème par exemple que deux graphes G_1 et G_2 ne sont pas isomorphes : quand G_1 et G_2 ne sont pas isomorphes, il est possible pour P de convaincre V d'accepter, si les deux graphes sont isomorphes, même un prouveur malhonnête ne peut pas persuader V d'accepter avec probabilité plus grande que $1/2$. Enfin, on appelle IP la classe des langages qui ont un système de preuve.

3. Montrer que $GNI \in IP$.

On peut montrer, mais c'est un problème difficile, que $PSPACE = IP$.

`Exercice 2` `Machine de Turing à oracle`

Un oracle A est un langage. Une machine de Turing à oracle M^A est une MT ordinaire avec une bande spéciale, appelée bande de l'oracle. Quand M écrit une chaîne w sur la bande de l'oracle, la machine apprend si w appartient à A en une étape de calcul.

On note P^A la classe des langages décidables par une MT à oracle en temps polynomial qui utilise l'oracle A. On définit de manière similaire NP^A. Par exemple, $NP \in P^{SAT}$ et de même $co - NP \in P^{SAT}$ car P^{SAT} est clos par complémentation.

1. Deux formules booléennes sur le même ensemble de variables sont équivalentes si elles ont la même valeur pour tout assignation des variables. On dit qu'une formule booléenne est minimale s'il n'y a pas de formule équivalente de taille plus petite.

Soit $NONMIN - FORMULA = \{\langle\phi\rangle \mid \phi \text{ n'est pas une formule booléenne minimale}\}$.

Montrer que $NONMIN - FORMULA \in NP^{SAT}$. (Il ne semble pas que $NONMIN - FORMULA$ appartienne à NP.)

2. Montrer qu'il existe un oracle A tel que $P^A = NP^A$.

3. Montrer qu'il existe un oracle B tel que $P^B \neq NP^B$. Pour ce faire, on pourra étudier le langage $L_B = \{w \mid \exists x \in B,\ |w| = |x|\}$ et construire un oracle B tel que L_B n'est décidé par aucune machine en temps polynomiale avec l'oracle B.

`Exercice 3` `Un problème EXPSPACE-complet`

On rappelle que $EXPSPACE = \bigcup_{k\in\mathbb{N}} SPACE(2^{n^k})$. On considère une généralisation des expressions régulières, c'est-à-dire des expressions qui sont construites à partir de $\emptyset$, ε et des éléments d'un alphabet A en utilisant les opérations rationnelles : union ($\cup$), concaténation (.), itération (*) et exponentiation ($\uparrow$). Si R est une expression régulière et k un entier positif, l'écriture $R \uparrow k$ est équivalente à la concaténation de R k fois. On peut aussi écrire R^k pour $R \uparrow k$. Il est clair que les expressions régulières et les expressions régulières généralisées génèrent la même classe des langages rationnels car on peut éliminer une exponentiation en utilisant plusieurs concaténations.

1. Donner un algorithme non-déterministe en espace linéaire qui décide le langage $EQ_{NFA} = \{\langle M_1, M_2\rangle : M_1,\ M_2$ sont des automates finis non-déterministes et $L(M_1) \neq L(M_2)\}$ où A est l'alphabet d'entrée.

2. Montrer que le problème de tester l'équivalence de deux expressions régulières sans opérateur exponentiation est résoluble en espace polynomial, soit

$$EQ_{REX} = \{\langle R, S\rangle : R \text{ et } S \text{ sont des expressions régulieres équivalentes}\} \in PSPACE.$$

3. Montrer que

$$EQ_{REX\uparrow} = \{\langle R, S\rangle \ :\ R \text{ et } S \text{ sont des expressions régulieres équivalentes}\}$$

est EXPSPACE-complet.