TD 4 – Machines de Turing, hiérachie en temps, temps polynomial

Exercice 1.

1. Décrire une machine de Turing reconnaissant le langage
$$\{m \in \{a,b\}^* \mid \ m \text{ contient autant de } a \text{ que de } b\}$$
2. Décrire une machine de Turing reconnaissant l'ensemble des palindromes sur $\{a,b\}$.

Exercice 2. Montrer qu'une machine de Turing à $k > 1$ rubans de travail fonctionnant en temps t peut être simulée par une machine de Turing à un seul ruban de travail fonctionnant en temps $O(t^2)$. D'où vient la perte de temps ?

Exercice 3. Expliquer comment il est possible de simuler efficacement le calcul d'une machine de Turing utilisant des rubans bi-infinis sur une machine de Turing utilisant des rubans semi-infinis.

Exercice 4.

1. Montrer que si, pour tout $n \in \mathbb{N}$, $f(n) \leqslant g(n)$, alors $\mathrm{DTIME}(f(n)) \subseteq \mathrm{DTIME}(g(n))$.
2. Montrer que s'il existe $N \in \mathbb{N}$ tel que, pour tout $n \geqslant N$, $f(n) \leqslant g(n)$ et si, pour tout $n \in \mathbb{N}$, $g(n) \neq 0$, alors $\mathrm{DTIME}(f(n)) \subseteq \mathrm{DTIME}(g(n))$.
3. Montrer que si, pour tout n, $t(n) \geqslant n$, alors $\mathrm{DTIME}(t(n))$ est clos par union finie, intersection finie et complémentaire.

Exercice 5. Montrer que les fonctions ci-dessous sont constructibles en temps :

1. $t(n) = c$,
2. $t(n) = n$,
3. $t(n) = 2^{n^c}$,
4. Montrer que si t_1 et t_2 sont constructibles en temps, alors il en est de même pour $t_1 + t_2$ et $t_1 t_2$.
5. Montrer qu'une fonction f constructible en temps et telle que $f(n) = o(n)$ est ultimement constante.

Exercice 6. Théorème de la lacune (Trakhtenbrot).
Le but de cet exercice est de montrer qu'il existe une fonction calculable $f : \mathbb{N} \to \mathbb{N}$ telle que $f(n) \geqslant n$ et $\mathrm{DTIME}(f(n)) = \mathrm{DTIME}(2^{f(n)})$.

1. Comparer avec le théorème de hiérarchie en temps vu en cours.
2. On considère $(M_i)_{i\in\mathbb{N}}$ une énumération des machines de Turing. Montrer qu'il suffit de construire une fonction $f : \mathbb{N} \to \mathbb{N}$ vérifiant la propriété suivante : pour tout i, aucune des machines $M_0, \ldots, M_i$ ne s'arrête sur aucune entrée de taille i en un temps compris entre $f(i)$ et $i2^{f(i)}$.
3. Construire une fonction calculable f qui vérifie la propriété ci-dessus. Conclure.
4. Existe-t-il une fonction f calculable telle que $\mathrm{DTIME}(f(n)) = \mathrm{DTIME}\left(2^{2^{2^{f(n)}}}\right)$?

Exercice 7. Montrer que les problèmes suivants sont dans P.

1. L'ensemble des graphes connexes.
2. L'ensemble des arbres.

Exercice 8. On considère le langage $L = \{1^n \mid n \text{ premier}\}$, i.e. l'ensemble des mots qui sont une suite de 1 dont la longueur est un nombre premier. Montrer que L appartient à P.

Exercice 9. Montrer que les problèmes suivants sont dans P.

1. DNF-Sat, l'ensemble des (codes de) formules propositionnelles sous forme normale disjonctive qui sont satisfaisables. (Rappel : une formule propositionnelle est sous forme normale disjonctive si elle est de la forme $\bigvee_i \bigwedge_j \ell_{i,j}$, où les $\ell_{i,j}$ sont des variables ou des négations de variables).
2. CNF-Taut, l'ensemble des (codes de) formules propositionnelles sous forme normale conjonctive qui sont tout le temps vraies (quelle que soit l'affectation).

Exercice 10. Clauses de Horn.

1. Une clause de Horn est une disjonction de littéraux contenant au plus un littéral positif (par exemple $\neg x \vee \neg y \vee z$, ou x, ou $\neg x \vee \neg y$, mais pas $x \vee y$). Le problème HornSat est de déterminer, sur la donnée d'un ensemble de clauses de Horn, si leur conjonction est satisfaisable. Montrer que HornSat est dans P. (On pourra considérer l'opération suivante : trouver une clause réduite à un seul littéral, celui-ci doit alors prendre la valeur vraie, et on peut donc modifier les autres clauses qui contiendraient ce littéral ou sa négation. Étudier ce qui arrive lorsqu'on itère cette opération.)
2. DualHornSat est le problème de déterminer la satisfaisabilité d'un ensemble de clauses où chaque clause contient au plus un littéral négatif. Montrer que DualHornSat est dans P.

Exercice 11. Soit Det le problème de déterminer, sur la donnée d'une matrice carré d'entiers A et d'un entier i, la valeur du i-ième bit de $\det(A)$.

1. Si on a une borne n sur les dimensions de A et m sur le nombre de bits de chaque coefficient entier de la matrice A, donner une borne sur le nombre de bits de $\det(A)$ qui soit polynomiale en n et m.
2. Expliquer pourquoi le problème Det appartient à P.

Un graphe biparti est un graphe $G = (V_1 \dot\cup V_2, E)$, où E est un sous-ensemble de $V_1 \times V_2$ appelé l'ensemble des arcs de G. Supposons que $V_1 = V_2 = \{1, \dots, n\}$. On associe à G une matrice A de dimention $n \times n$:

$$A_{i,j} = \begin{cases} x_{i,j} & \text{si } (i,j) \in E \\ 0 & \text{sinon} \end{cases}$$

où les $x_{i,j}$ sont des variables. Le déterminant de A est alors un polynôme en les variables $x_{i,j}$. Un couplage parfait de G est un sous-ensemble C de E tel que tout sommet de $V_1 \dot\cup V_2$ appartienne à exactement un seul arc de C.

3. Montrer que G a au moins un couplage parfait ssi $\det A$ n'est pas le polynôme identiquement nul.
4. Peut-on en déduire un algorithme fonctionnant en temps polynomial et déterminant si un graphe biparti a un couplage parfait ?

Exercice 12.

Soit X un problème dans P et Y un problème non trivial (i.e. non vide et différent de Σ^* tout entier). Montrer que X se réduit à Y pour la réduction many-one en temps polynomial.