TD03 - Déterminisme or not determinisme

Exercice 1. *Ladner's Theorem*

Dans la suite, il est supposé que P $\neq$ NP. Nous allons construire un langage $A \in$ NP qui n'est ni dans P, ni NP-complet.

Rappels :

– Machine de Turing à horloge : machine de Turing disposant d'un compteur de transitions, et terminant au bout d'un nombre d'étapes déterminé.
– Pour une machine de Turing M calculant la fonction f, et pour un langage X on notera $L_X(M)$ le langage des mots x tels que $f(x) \in X$. Il s'agit donc du langage qui se réduit à X par la fonction de transition f.

1. Montrer que grâce aux horloges on peut construire une énumération (M_i) de machines de Turing s'arrêtant au bout d'un temps polynomial et telle que $P = \bigcup L(M_i)$. Montrer que l'on peut caractériser de même NP à l'aide d'une énumération de machines de Turing (M'_i) avec oracle s'arrêtant en temps polynomial.

2. Donner des conditions suffisantes sur A et les M_i et M'_i pour que A ne soit ni polynomial, ni NP-complet.

3. Cherchons un A de la forme $SAT \cap S$ où SAT est le problème bien connu et $S = \bigcup_{i=0}^{\infty} [n_{2i}, n_{2i+1}[$ l'ensemble des mots de longueur comprise entre n_{2i} et n_{2i+1}.
Définir (n_i) tel que les conditions de la question précédente soient vérifiées.

4. Vérifier que l'on a : $S \in P$. A est donc bien dans NP.

Exercice 2. *Retour sur les simulations universelles*

1. Montrer que l'on peut construire une machine de Turing non déterministe universelle pour les machines non déterministes, et fonctionnant en temps $O(n)$ sur les mots de taille n.
(Chercher une solution plus simple que la semaine dernière)