Indécidabilité, Réductions 2

Exercice 1 (Quelques réductions classiques)
Montrer que les problèmes suivants sont indécidables :
Étant donné un code $< M > \in \{0,1\}^*$ d'une machine de Turing, décider si :
- M s'arrête sur le mot vide.
- M s'arrête pour au moins une entrée.
- M s'arrête sur toutes ses entrées.

Exercice 2 (Automates linéairement bornés)
Un *automate linéairement borné* est une machine de Turing dont l'espace de travail est limité à droite par l'extrémité du mot d'entrée. On considère les problèmes suivants :

Donnée : Un automate linéairement borné M, et un mot w.

Question : Est-ce que w est accepté par M ?

Donnée : Un automate linéairement borné M.

Question : Est-ce que le langage de M est vide ?

Donnée : Un automate linéairement borné M.

Question : Est-ce que, pour tout w, M s'arrête sur w ?

Étudier la décidabilité de chacun de ces problèmes.

Exercice 3 (Théorème du point fixe)
On considère un alphabet Σ. Étant donné $u \in \Sigma^*$, on note M_u la machine dont le code dans Σ^* est u. Nous allons prouver le théorème suivant :

Théorème 1 *Soit $f : \Sigma^* \to \Sigma^*$ une fonction totale calculable, alors il existe un mot $u \in \Sigma^*$ tel que*

$$L(M_u) = L(M_{f(u)})$$

1. On considère une telle fonction f, calculée par la machine K. Soit N la machine de Turing qui, sur l'entrée x, calcule la description de la machine de Turing suivante :
   (a) Prend en entrée y,
   (b) Construit la machine M_x,
   (c) Simule M_x sur l'entrée x,
   (d) Si cette simulation s'arrête en produisant le mot $M_x(x)$, simule K sur ce mot,
   (e) Interprète ce résultat $K(M_x(x))$ comme le code d'une machine de Turing, et simule cette machine de Turing sur l'entrée y. Accepte ou rejette selon que cette machine accepte ou rejette.

Montrer que N est totale, et que l'identité suivante est satisfaite :

$$L(M_{N(x)}) = L(M_{f(M_x(x))})$$

2. Soit $v \in \Sigma^*$ tel que $N = M_v$. Montrer que $N(v)$ est le point fixe recherché.

Exercice 4 (Problème de la convergence d'une suite d'éléments de $\mathbb{Z}$)
Définition 1 *On dit qu'une suite est définie récursivement si pour tout entier $n \in \mathbb{N}^*$, on peut calculer effectivement son n-ème terme.*
On appelle *problème de la convergence d'une suite d'éléments de* $\mathbb{Z}$ le problème suivant :

Donnée : une suite d'éléments de $\mathbb{Z}$ définie récursivement

Question : cette suite est-elle convergente ?

Montrer que ce problème est indécidable.

Exercice 5 (Problème du zéro dans les matrices 3×3)
On appelle *problème du zéro dans les matrices* 3×3 le problème suivant :

Donnée : un ensemble fini de matrices 3×3 à coefficients dans $\mathbb{Z}$

Question : existe-t-il un produit fini $M = M_{i_1}.M_{i_2} \dots M_{i_r}$ tel que l'élément $(3, 2)$ de M soit nul ?

Supposons l'alphabet Σ numéroté de 1 à n, et notons b la fonction associant à un mot $u \in \Sigma^*$ sa valeur en le considérant comme un codage en base $n + 1$ en prenant sa première lettre comme le bit de poids faible. Notons également ℓ la fonction qui associe à $u \in \Sigma^*$ l'entier $(n + 1)^{|}u|$ où $|u|$ dénote la longueur de u.
En considérant l'application φ de $\Sigma^* \times \Sigma^*$ dans l'ensemble des matrices 3×3 suivante, montrer que le problème du zéro dans les matrices 3×3 est indécidable.

$$\varphi(u, v) = \begin{pmatrix} \ell(u) & \ell(u) - \ell(v) & 0 \\ 0 & \ell(v) & 0 \\ b(v) & b(v) - b(u) & 1 \end{pmatrix}$$

Exercice 6 (Énumération de langages)
Définition 2 *Soit Σ un alphabet fini.*
Une machine de Turing avec canal de sortie *est une machine de Turing ne prenant pas d'entrée, qui possède un ruban supplémentaire, sur lequel l'écriture et le déplacement de la tête de lecture se font toujours vers la droite, et qui possède un séparateur* \$*. Le* langage énuméré par M *est défini par $G(M) = \{w \in \Sigma^* \mid w$ est imprimé entre deux symboles \$$\}$.*

- Montrer qu'un langage L est récursivement énumérable si, et seulement si, il est énuméré par une machine de Turing avec canal de sortie.
- Montrer qu'un langage L est récursif si, et seulement si, il peut être énuméré par une machine de Turing avec canal de sortie en ordre lexicographique et par longueur croissante.

Exercice 7 (Hiérarchie arithmétique)
Définition 3 *Une machine de Turing à oracle B est une machine de Turing classique, disposant en plus d'un ruban infini sur lequel est écrit la réponse, pour tout mot w, au problème $w \in B$?. Cette machine peut donc, en un temps fini, obtenir la réponse à la question d'appartenance pour le problème B.*
On dit qu'un problème A est récursivement éunumérable dans B *s'il existe une machine de Turing M avec oracle B qui l'accepte ($A = L(M)$). De plus si M s'arrête sur toutes ses entrées, on dit que A est* récursif dans B.

On peut alors considérer les ensemble suivants :

$$\begin{array}{rcl}
\Sigma_1 & = & RE \\
\Delta_1 & = & R \\
\Pi_1 & & co - RE \\
\Sigma_{n+1} & = & \{\text{ensembles r.e. dans un certain } B \in \Sigma_n\} \\
\Delta_{n+1} & = & \{\text{ensembles récursifs dans un certain } B \in \Sigma_n\} \\
\Pi_{n+1} & = & \{\text{complémentaires des ensembles appartenant à } \Sigma_{n+1}\}
\end{array}$$

1. Montrer que, pour tout $n \geq 1$, on a $\Delta_n = \Sigma_n \cap \Pi_n$.
2. Donner, pour tout $n \geq 1$, un langage appartenant à Σ_n et pas à Π_n. Autrement dit : montrer que la hiérarchie arithmétique est stricte.
   Indication : généraliser la construction d'un langage récursivement énumérable et pas récursif.
3. Placer dans la hiérarchie les langages suivants :
   – LU, $\overline{LU}$, HP, $\overline{HP}$
   – $EMPTY = \{< M >| L(M)$ est vide $\}$
   – $TOTAL = \{< M >| L(M)$ est total $\}$
   – $FINI = \{< M >| L(M)$ est fini $\}$
   – $COFINI = \{< M >| \overline{L(M)}$ est fini $\}$