Exercice 0 :

(Rappel) Montrer que L est un langage récursivement énumérable, si et seulement s'il existe une machine de Turing M à deux rubans et un état q_e de M tels que,

$(\epsilon, q_0, \$, \$) \vdash_M^* (\epsilon, q_e, \$w', \$w)$ si et seulement si $w \in L$.

Exercice 1 :

Montrer que, si f est calculable et L est récursivement énumérable, alors $f(L)$ est récursivement énumérable.

Exercice 2 :

.

1. Donner explicitement la table d'une machine de Turing qui, étant donné un mot $w \in \{0,1\}^*$, accepte w si w contient au moins autant de 0 que de 1 et rejette sinon. (On prendra soin de démontrer que la machine fait bien ce qu'elle est censée faire).
2. Même question avec "strictement plus" au lieu de "au moins autant".

Exercice 3 :

La fonction suivante est-elle calculable ? Pour tout mot sur l'alphabet $\{a, b, c, d, e\}$, on renvoie 1 si le nombre de c est compris (au sens strict) entre le nombre de a et le nombre de b, et on renvoie 0 sinon.

Exercice 4 :

On a vu dans le cours que si f est **semi-calculable** et L est r.e., alors $f^{-1}(L)$ est r.e. Maintenant, si f est **calculable** et L est r.e., a-t-on $f^{-1}(L)$ est décidable.

Exercice 5 :

Montrer que, si f est semi-calculable et L est récursivement énumérable, alors $f(L)$ est récursivement énumérable.

Exercice 6 :

On considère le modèle de calcul des machines de Turing à ruban bi-infini : la définition est la même que la définition, excepté qu'il n'y a pas de symbole spécial \$ (et donc pas d'hypothèse sur les règles correspondantes). Une configuration initiale est un triplet (ϵ, q_0, wB). Une configuration finale est un triplet (w, q, w') avec $q \in \{\mathbf{accept}, \mathbf{reject}\}$. La relation de transition est donnée comme dans la définition, excepté que, lors d'un mouvement gauche, $w_1 = B$ lorsque $w = c$.

La définition de fonction calculable est la même que pour les machines de Turing avec un ruban infini d'un seul côté : on ignore les blancs dans le résultat.

Montrer qu'une fonction est calculable dans ce modèle si et seulement si elle est calculable par une machine de Turing.

Exercice 7 :

Une machine de Turing avec états finaux (MTF) est presque comme une MT du cours, mais il n'y a pas d'état acceptant ou rejetant. Un mot en entrée est accepté s'il existe un calcul maximal qui termine dans un état final, rejeté sinon. Un langage est MTF-semi-decidable s'il existe une MTF qui accepte exactement ses mots. Si en plus cette MTF ne permet de calcul infini sur aucune entrée, le langage est MTF-décidable. La fonction associée à une MTF a pour domaine le langage reconnu et pour valeur le plus grand préfixe sans blanc sur le ruban au moment de l'arrêt. Cette fonction est dite MTF-sem-calculable. Si sont domaine est MTF-décidable, elle est dite MTF-calculable.

1. Un langage est semi-décidable ssi il est MTF-semi-décidable.
2. Un langage est décidable ssi il est MTF-décidable.
3. Une fonction est semi-calculable ssi elle est MTF-semi-calculable.
4. Une fonction est calculable ssi elle est MTF-calculable.

Exercice 8 :

Un automate à deux piles est un tuple $(Q, q_0, \Sigma, \delta, \perp)$, où $\perp \in \Sigma$ et $Q = Q_1 \uplus Q_2$ et $\delta :\subseteq Q \times \Sigma \to Q \times \Sigma^*$ est une fonction partielle.

Une configuration est un triplet (q, w, w') où $w, w' \in \Sigma^*$. Une configuration de départ est de la forme $(q_0, \perp, w\perp)$, où $w \in (\Sigma \setminus \perp)^*$ est appelé le mot d'entrée.

Une étape de calcul est de la forme $(q, aw, w') \vdash (q', \alpha w, w')$ si $q \in Q_1$ et $\delta(q, a) = (q', \alpha)$ ou de la forme $(q, w, bw') \vdash (q', w, \beta w')$ si $q \in Q_2$ et $\delta(q, b) = (q', \beta)$.

Le mot w est reconnu par l'automate si en partant de $(q_0, \perp, w\perp)$ on arrive à une configuration de la forme (q, ϵ, w'), i.e. la première pile est vide.

Un langage est ADP-semi-decidable si ses mots sont reconnu par un ADP ; ADP-decidable si en plus il n'y a pas de calcul infini. Pareil pour les fonctions, dont la valeur est donnée par la deuxième pile au moment ou la première se vide.

Montrer l'équivalence avec les langages et les fonctions (semi)-calculables.