TD 11

Exercice 1. *Mangez des réductions*

On fixe un alphabet fini Σ contenant entre autre les lettres a et b. ϵ désignera le mot vide. Posons Σ^* l'ensemble des mots finis sur Σ. On notera enfin M_σ avec $\sigma \in \Sigma^*$ la machine de Turing effectuant le programme de code σ. (Plus formellement, $M_\sigma(x) = U(<\sigma, x>)$)

✎ Les ensembles suivants sont-ils récursifs ? récursivement énumérables ? de complémentaire récursivement énumérable ?

1. $\{\sigma \in \Sigma^*, M_\sigma(\sigma)$ s'arrête$\}$
2. $\{\sigma \in \Sigma^*, M_\sigma(\epsilon)$ s'arrête$\}$
3. $\{\sigma \in \Sigma^*, M_\sigma(ab)^\frown M_\sigma(ba) = aaa\}$ (avec $^\frown$ l'opérateur de concaténation)
4. $\{\sigma \in \Sigma^*, \forall x, M_\sigma(x) = x$ si $M_x(x)$ s'arrête et b sinon$\}$, avec $b \notin \Sigma$.
5. $\{\sigma \in \Sigma^*, \exists x, M_\sigma(x)$ ne s'arrête pas$\}$
6. $\{\sigma \in \Sigma^*, \exists w \in \Sigma^*, M_w(\sigma) = abb\}$
7. $\{\sigma \in \Sigma^*, M_\sigma$ ne s'arrête sur aucun mot dont σ est un préfixe$\}$

Exercice 2. *Algébriques*

1. Montrez que n'importe quel langage algébrique peut être reconnu par une machine de Turing.
2. Construisez une machine de Turing qui reconnaît le langage $\{a^n b^n c^n | n \in \mathbb{N}\}$.
3. On considère un nouveau modèle de machine de Turing ne pouvant que aller à droite (ou ne pas se déplacer) sur le ruban. Plus formellement, sa fonction de transition est de la forme
$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{R, S\}.$$
Ce modèle est-il équivalent au modèle standard ? Si oui, le prouver. Si non, quelle classe de langages reconnaît-il exactement ?

Exercice 3. *Canard analphabète*

On dit qu'une machine **énumère** un langage L si elle écrit tous les $w \in L$ sur son ruban. (Si L est infini la machine ne termine pas). On rappelle qu'un langage est récursivement énumérable s'il existe une machine de Turing qui accepte exactement les mots du langage.

1. Montrer qu'un langage est récursivement énumérable si et seulement si il existe une machine de Turing qui l'énumère.
2. Montrer que s'il existe une machine de Turing qui énumère L dans l'ordre lexicographique, alors L est récursif.
3. Soit L un langage récursivement énumérable infini. Montrer qu'il existe un sous-ensemble infini de L qui est récursif.

Exercice 4. *Moins de place*

Une machine de Turing est dite *linéairement bornée* si elle n'écrit pas en dehors de l'espace utilisé par la donnée.
Comme on peut toujours supposer qu'une machine de Turing n'écrit pas le symbole blanc $\sqcup$ (en le dupliquant éventuellement en un autre symbole $\sqcup_2$), on peut aussi dire qu'une machine linéairement bornée est une machine telle que si $\delta(p, \sqcup) \ni (q, b, x)$ est une transition alors $b = \sqcup$ et $x = \leftarrow$. Ceci empêche la machine de modifier les symboles blancs présents sur la bande.

1. Etant donné un mot w et une machine linéairement bornée M, peut-on décider si M accepte w ?
2. Etant donné une machine linéairement bornée M, peut-on décider si M n'accepte aucun mot, c'est-à-dire $L(M) = \emptyset$?

   Indice : On pourra commencer par montrer que ce problème est indécidable pour les machines de Turing en général.

Exercice 5. *Sempervivum gleyifié*

Définition 1. *Soit M une machine de Turing sur l'alphabet Σ travaillant sur un seul ruban.*
*On dit qu'une transition $\delta(q, a)$ est **pivotale** si elle change l'état interne de la machine, i.e. si on a $\delta(q, a) = (q', b, F)$ avec $q \neq q'$.*
*On note **piv**(M, x) le nombre de transitions qui ont été pivotales lors de l'exécution de la machine M sur l'entrée x. On définit la **complexité pivotale** d'une machine de Turing comme étant la fonction $piv_M : \mathbb{N} \to \mathbb{N}$ telle que $piv_M(n) = \max_{x \in \Sigma^n} piv(M, x)$.*
*On appelle **PIV**$(f(n))$ la classe des langages reconnaissables par une machine de Turing de complexité pivotale $O(f(n))$.*

1. Donnez une majoration de la complexité pivotale d'une Machine de Turing qui travaille en temps $O(f(n))$.
2. Trouvez f telle que les langages réguliers soient dans $PIV(f(n))$.
3. Montrez que si on accepte de travailler sur deux rubans, il existe une machine de Turing universelle de complexité pivotale constante.

Colors weave into a Spire of Flame...