Devoir de Calculabilité

À rendre le 10 octobre 2016 au plus tard

Dans tout l'énoncé, les machines de Turing sont supposées disposer de deux rubans, le deuxième ruban étant dit "ruban de sortie" (même si aucune hypothèse n'est effectuée sur la lecture/écriture sur ce ruban). Dans ce contexte, la donnée de la machine de Turing sera toujours sur le premier ruban: la configuration initiale de M sur w est $(q_0, (\epsilon, \$w), (\epsilon, \$)$. Le résultat du calcul de M sur w (si M s'arrête sur w) sera le contenu du deuxième ruban (auquel on enlève le préfixe \$ et les blancs en fin de mot).

Question 1 (un théorème de point fixe)

On note M_U la machine universelle.

Soit f est une fonction calculable qui envoie les codes de machines de Turing sur des codes de machine de Turing (et renvoie le mot vide sur les entrées qui ne sont pas des codes de machines de Turing).

Montrer qu'il existe une machine M telle que, sur toute donnée x,

$$M_U(f(< M >), < x >) =< M(x) > .$$

Ind: considérer la fonction g qui à $< M >$ associe le code de la machine $M(M)$ qui à x associe $M(< M >)(x)$ quand $M(< M >)$ est le code d'une machine de Turing et ne termine pas sinon. Considérer ensuite $g(< M_{f \circ g} >)$.

Question 2

Dans cette partie, on appellera *virus* une machine de Turing M comportant deux rubans et un état spécial q_{emit}, telle que, sur la donnée ϵ, le calcul de M passe par une configuration $(q_{\text{emit}}, (w_1, w_2), (\epsilon, \$ < M >))$.

1. Montrer qu'il existe des virus. (On pourra utiliser la question 1)

2. Montrer qu'avec cette définition, il est possible de détecter les virus: il existe une machine M_d telle que, sur la donnée $< M >$, M_d s'arrête en acceptant si et seulement si M est un virus.

3. Montrer qu'en revanche l'ensemble des virus n'est pas récursif.

Question 3

L'ensemble $\mathcal{V}$ est *machines de Turing infectées par un virus* est le plus petit ensemble de (codes de) machines de Turing tel que:

- Les virus appartiennent à $\mathcal{V}$
- Si le calcul de M sur ϵ passe par une configuration dans laquelle le ruban de sortie contient le code d'une machine de Turing $M' \in \mathcal{V}$, alors $M \in \mathcal{V}$.

Montrer que $\mathcal{V}$ est récursivement énumérable et pas récursif.

Question 4

Dans cette partie, on appellera *virus* une machine de Turing M comportant un ruban de sortie tel que, sur *toute donnée*, le calcul de M passe par une configuration dans laquelle le ruban de sortie contient $< M >$.

La définition des machines infectées par un virus est la même que dans la question précédente, avec cette nouvelle définition de virus.

Montrer qu'alors l'ensemble des machines infectées par un virus n'est ni récursivement énumérable, ni co-récrusivement énumérable.

Question 5

On revient à la définition de virus de la question 2. Mais il est plus réaliste de considérer qu'un virus ne se reproduit pas nécessairement à l'identique, mais produit un clone qui est fonctionnellement identique.

Soit $\sim$ l'équivalence observationelle: étant données deux machines de Turing V, V', $V \sim V'$ si, pour toute donnée x la séquence $w_1, \ldots, w_n, \ldots$ des contenus du deuxième ruban quand V est dans l'état q_{emit} est la même que la séquence $w'_1, \ldots, w'_n$ des contenus du deuxième ruban quand V' est dans l'état q_{emit}.

Montrer que l'ensemble des machines M qui sont équivalentes à un virus n'est ni récursivement énumérable, ni co-récursivement énumérable. Autrement dit, ce type de virus est indétectable.