Modèles de Calcul
Machines de Turing
Fiche 3 - Calculabilité

L3-Informatique 2021

Exercice 1

On considère la fonction $f(x) = \varphi_x(x) + 1$. Montrer que

1. f est Turing calculable partielle.
2. Il n'existe pas de fonction Turing calculable totale g qui prolonge f, c'est à dire telle que:
$$g = \begin{cases} f(x) & \text{si } f(x)\downarrow \\ v & \text{si } f(x)\uparrow \end{cases}$$
3. Montrer qu'il n'existe pas d'énumération effective des fonctions décidables, c'est à dire une liste $h_1(x), h_2(x), \ldots$ telle que
   - Pour tout i on a $h_i \in R$
   - Pour toute fonction $h \in R$ alors il existe un i tel que $h = h_i$.
   - La fonction $V(i, x) = h_i x$ est calculable.
4. Pourquoi une telle énumération est quand même possible pour les fonctions de RE ?

Exercice 2

On suppose pour les trois premières questions que le prédicat A se réduit à B, c'est à dire qu'il existe une fonction f totale et calculable telle que $A(x) = 1$ si et seulement si $B(f(x)) = 1$. On note $\mathcal{A}$ et $\mathcal{B}$ les ensembles suivants :
$$\mathcal{A} = \{x \mid A(x) = 1\} \qquad \mathcal{B} = \{x \mid B(x) = 1\}$$
autrement dit les prédicats A, B sont les fonctions caractéristiques des ensembles $\mathcal{A}, \mathcal{B}$.

- Montrez que si $\mathcal{B}$ est décidable alors $\mathcal{A}$ l'est également.
- Montrez que si $\mathcal{B}$ est récursivement énumérable alors $\mathcal{A}$ l'est également.
- Expliquez comment formuler les résultats précédents lorsqu'on veut prouver qu'un ensemble est indécidable, ou non récursivement énumérable. (La question est volontairement vague, la réponse est très courte et vous servira pour résoudre les trois dernières questions de cet exercice).

On considère la procédure suivante :

```
int gq(int p (int),int x, int n)
{
   int y = p(x);
   return n;
}
```

On note $q = f(p, x)$ la fonction définie par $q(n) = gq(p, x, n)$. On note $q = f(p, x)$ la fonction définie par $q(n) = gq(p, x, n)$.

- Montrez que f réduit le problème de la *terminaison* du calcul de $p(x)$ à celui de savoir si q calcule l'identité.
- En déduire que l'ensemble des procédures qui *ne calculent pas l'identité* n'est pas récursivement énumérable.

On remplace la procédure gq par une procédure gr, où la variable t représente le temps ; On suppose l'existence d'un prédicat h total et calculable tel que $h(p, x, t) = 1$ si et seulement si $p(x)$ termine en un temps plus petit que t.

```
int gr(int p (int), int x, int t)
{
  if (h(p,x,t)) return p(x)
     else return t;
}
```

On note $r = g(p, x)$ la fonction définie par $r(t) = gr(p, x, t)$.

- Montrer que g réduit le problème de la terminaison du calcul de $p(x)$ à celui de savoir si r *ne calcule pas* l'identité.
- En déduire que l'ensemble des procédures qui calculent l'identité n'est pas récursivement énumérable.

Exercice 3

On note Λ_x l'ensemble d'entiers dont la fonction caractéristique est codée par la machine de Turing numéro x. On considère que si $\varphi_x(y)$ converge et donne comme résultat un entier différent de 0 alors $y \in \Lambda_x$.

On définit les ensembles suivants :

- $E_v = \{x|\ v \in \Lambda_x\}$.
- $E_{ne} = \{x|\ \Lambda_x \neq \emptyset\}$.
- $E_e = \{x|\ \Lambda_x = \emptyset\}$.
- $E_k = \{x|\ |\Lambda_x| \geq k\}$.
- $E_{\mathcal{N}} = \{x|\ \Lambda_x = \mathbb{N}\}$.
- $E_f = \{x|\ \Lambda_x$ est fini$\}$.
- $E_r = \{x|\ \Lambda_x \in T - calculable\ totale\}$.
- $E_{nr} = \{x|\ \Lambda_x \notin T - calculable\ totale\}$.

Montrer que les assertions suivantes sont vraies :

1. Aucun des ensembles ci-dessus n'est récursif.
2. $E_v \in RE$.
3. $E_{ne} \in RE$.
4. $E_e \notin RE$.
5. $E_k \in RE$.
6. $E_r \notin RE$.
7. $E_{nr} \notin RE$.