Magistère STIC
Calculabilité
Devoir à rendre avant le 4 novembre 2004

22 octobre 2004

L'objet du problème est l'étude de divers modèles de calcul, en montrant que ces modèles ont le même pouvoir d'expression que les machines de Turing. En annexe on trouvera les définitions d'automates à pile(s), machines à compteurs, système d'addition de vecteurs, jeux rationnels...

Question 1

Montrer que le problème suivant est indécidable :

Donnée : le code d'une machine de Turing M

Question : est-ce que $L(M) = \emptyset$?

Question 2. (Automates à piles)

1. Montrer que, si L est un langage récursivement énumérable, alors il existe un automate A à deux piles tel que $L(A) = L$.
2. En déduire que le problème de savoir, étant donné un automate A à 2 piles, si $L(A)$ est vide, est indécidable.
3. **Question facultative**. Montrer que si A est un automate à une pile, alors $L(A)$ est récursif .

Question 3. (Machines de Minsky)

1. Soit $k > 1$. Montrer qu'on peut construire une machine de Minsky M_k telle que, sur la donnée $n \in \mathbb{N}$, M_k s'arrête dans une configuration où le premier registre contient q et le deuxième registre contient r où q, r sont respectivement le quotient et le reste de la division euclidienne de n par k.
2. Soit $\Sigma \setminus \{\$, B\} = \{a_1, \ldots, a_k\}$. Pour $w \in (\Sigma \setminus \{\$, B\})^*$, on note $c_k(w)$ l'entier dont l'écriture en base $k+1$ est w.

   Montrer que, si L est récursivement énumérable, il existe une machine de Minsky à 4 registres qui accepte $\{c_k(w) \mid w \in L\}$. (On pourra utiliser la question 2).

3. Montrer le même résultat que celui de la question précédente, mais avec une machine à 2 registres (au lieu de 4). Ind : on pourra considérer le codage de 4 entiers en un seul par $\overline{(n_1, n_2, n_3, n_4)} = 2^{n_1} \times 3^{n_2} \times 5^{n_3} \times 7^{n_4}$.
4. Montrer que le problème de savoir si une machine de Minsky à 2 registres accepte au moins un entier est indécidable.

Question 4. (Systèmes d'addition de vecteurs)

Montrer que toute fonction (partielle) calculable (par une machine de Turing) sur les entiers est calculable par un système d'addition de vecteurs. En déduire que le problème :

Donnée : Un système d'addition de vecteurs S

Question : S accepte-t-il au moins un entier ?

est indécidable.

Question 5. (Jeux rationnels)

Montrer que toute fonction (partielle) calculable sur les entiers est calculable par un jeu rationnel.

A Automates à piles

Un automate à k piles $P_1, \ldots, P_k$ est donné par un ensemble fini d'états Q, un état initial q_0, un alphabet Σ contenant en particulier les symboles \$ et une fonction de transition $\delta : Q \times \Sigma^k \to (Q \cup \{\mathbf{accept}, \mathbf{reject}\}) \times (\Sigma \times \{\leftarrow, \to, \downarrow\})^k : \delta(q, a_1, \ldots, a_k) = (q', (a'_1, d_1), \ldots, (a'_k, d_k))$ telle que, si $a_i = \$$, alors $d_i \neq \leftarrow$.

Une configuration de la machine est constituée d'un état et de k mots non vides $P_1, \ldots, P_k \in \Sigma^*$. Un mouvement est défini par $q, P_1 a_1, \ldots, P_k a_k \vdash q', P'_1, \ldots, P'_k$ s'il existe une transition $q, a_1, \ldots, a_k \mapsto q', (a'_1, d_1), \ldots, (a'_k, d_k)$ telle que, pour tout $1 \leq i \leq k$,

1. Ou bien $d_i = \downarrow$ et $P'_i = P_i a'_i$
2. Ou bien $d_i = \to$ et $P'_i = P_i a_i a'_i$
3. Ou bien $d_i = \leftarrow$ et $P'_i = P_i$.

La configuration initiale de l'automate sur la donnée w est $q_0, \$\widetilde{w}, \$, \ldots, \$$ où $\widetilde{w}$ est l'image miroir de w (définie par $\widetilde{\epsilon} = \epsilon$ et $\widetilde{a \cdot w} = \widetilde{w} \cdot a$). Les définitions de calcul, langage accepté,... sont comme pour les machines de Turing.

B Machines de Minsky

Une telle machine est donnée par un ensemble fini d'états (contenant les états **accept** et **reject**), un état initial, N *registres* $r_1, \ldots, r_N$ et une fonction de transition de $Q \times \{0,1\}^N$ dans $Q \times \{+1, 0, -1\}^N : \delta(q, \alpha_1, \ldots, \alpha_N) = (q', d_1, \ldots, d_N)$ telle que si $\alpha_i = 0$, alors $d_i \neq -1$.

Une *configuration* de la machine est donnée par N entiers en base 1 (les contenus de registres) et un état.

Un mouvement de la machine est une relation entre configurations : $q, k_1, \ldots, k_N \vdash_M q', k'_1, \ldots, k'_N$ ssi il existe une transition de la machine $q, \alpha_1, \ldots, \alpha_N \mapsto q', d_1, \ldots, d_N$ telle que $\alpha_i = 0$ si $k_i = 0$ et $\alpha_i = 1$ sinon et, pour tout i, $k'_i = k_i + d_i$.

Un *calcul* de la machine sur la donnée n_0 est une suite de configurations, la première étant la configuration initiale $(q_0, n_0, 0 \ldots, 0)$ et telle que la $i$ème configuration est obtenue par un mouvement de la machine sur la $i-1$ème configuration.

Une machine de Minsky M *accepte* l'entier n_0, si le calcul de M sur n s'arrête en **accept**.

C Systèmes d'addition de vecteurs

Un tel système est donné par un entier n_0 et une suite finie de vecteurs $v_1, \ldots, v_n \in \mathbb{Z}^k$. Une configuration est un vecteur de $\mathbb{N}^k$. Le système peut faire un mouvement de la configuration v à la configuration v' s'il existe un entier $i \in \{1, \ldots, n\}$ tel que $v + v_1, \ldots, v + v_{i-1} \notin \mathbb{N}^k$ et $v + v_i = v' \in \mathbb{N}^k$.

Un *calcul* du système S sur la donnée $m \in \mathbb{N}$ est une suite de configurations $u_0 = (m, 0, \ldots, 0, n_0), u_1, \ldots, u_p, \ldots$ telle que S peut faire un mouvement de u_i à u_{i+1} pour toute paire d'indices $(i, i+1)$ de la séquence. Si la suite u_i est finie et qu'aucun mouvement ne permet de la prolonger, le calcul *s'arrête* et le résultat est la première composante de la dernière configuration.

Un système d'addition de vecteurs S *calcule* la fonction partielle $f : \mathbb{N} \to \mathbb{N}$ si, pour tout $m \in \mathbb{N}$,

- Ou bien $f(m)$ est défini et il existe un calcul de S sur m dont le résultat est $f(m)$.
- Ou bien $f(m)$ n'est pas défini et S ne s'arrête pas sur la donnée m.

Un système d'addition de vecteurs accepte un entier n si le calcul de S sur n s'arrête avec pour résultat 1.

D Jeux rationnels

Ces machines sont consituées d'une séquence finie de nombres rationnels $r_1, \ldots, r_n$ et d'un entier N. Une étape de calcul de la machine est définie comme suit : partant d'un entier naturel m, on multiplie m par le premier nombre rationnel de la séquence dont le dénominateur divise m. Lorsqu'on ne peut pas trouver de tel rationnel dans la séquence, la machine s'arrête.

On dit qu'un jeu rationnel $(r_1, \ldots, r_n), N$ *calcule* la fonction partielle f si, partant d'un nombre $2^k \times N$ et itérant les étapes de calcul,

- Ou bien le jeu s'arrête avec comme résultat $2^{f(k)}$ si $f(k)$ est défini
- Ou bien le jeu ne s'arrête pas sinon