Université Bordeaux 1
Master d'Informatique 1, 2009/2010

Modèles de Calcul : feuille 2

Modèles alternatifs aux machines de Turing .

Exercice 2.1

Pour chacune des fonctions suivantes de $\mathbb{N}^p$ dans $\mathbb{N}$, écrire une RAM qui la calcule :

1. $x \mapsto 2 * x$
2. $(x, y) \mapsto \mathrm{div}(x, y)$ (le quotient entier de x par y)
3. $(x, y) \mapsto \mathrm{r}(x, y)$ (le reste de la division entière de x par y)

Exercice 2.2

Une machine à k piles est définie par un triplet (Q, Σ, δ) où Q est un ensemble fini d'états, Σ est un alphabet fini et δ est un sous-ensemble fini de $(Q \times \Sigma \cup \{\perp\}) \times (Q \times (\Sigma^*)^k)$ avec $\perp \notin \Sigma$. La configuration d'une telle machine est donnée par $(q, w_1, \ldots, w_k)$ avec $q \in Q$, et $w_i \in \Sigma^*\perp$ pour tout i dans $[1; k]$. Elle peut passer d'une configuration $(q_1, w_1^1, \ldots, w_k^1)$ à une configuration $(q_2, w_1^2, \ldots, w_k^2)$ si $w_j^1 = X_j v_j^1$, $w_j^2 = u_j v_j^2$ et $((q_1, (X_1, \ldots, X_k)), (q_2, (u_1, \ldots, u_k))) \in \delta$[1].

1. Montrer que toute machine de Turing peut être simulée par une machine à deux piles.
2. Montrer que toute machine à k piles peut être simulée par une machine de Turing non-déterministe.

Exercice 2.3

1- Montrer que toute machine à deux piles peut être simulée par une RAM.
2- En déduire que toute machine de Turing peut être simulée par une RAM.

Exercice 2.4

Une machine à k compteurs est une machine qui utilise k registres r_1, ..., r_k qui contiennent chacun un entier. Un programme pour une telle machine est une série d'instructions numérotées de la forme :

1. n : $\textbf{inc}(r_i); \textbf{goto}\, m$
2. n : $\textbf{dec}(r_i); \textbf{goto}\, m$
3. n : $\textbf{if}\, r_i = 0\, \textbf{then goto}\, m_1 \textbf{else goto}\, m_2$
4. n : **stop**

[1]on suppose que si $X_k = \perp$ alors u_k appartient à $\Sigma^*\perp$.

où n, m, m_1 et m_2 dénotent des numéros d'instruction. On convient également que si $\mathbf{dec}(r_i)$ est exécutée alors que r_i contient zéro, la valeur de r_i reste zéro. La configuration d'une machine à k compteurs est donnée par un tuple $(n, p_1, \ldots, p_k)$ où n est le numéro de l'instruction courante et p_i est le nombre contenu dans le registre r_i. La transition d'une configuration à une autre s'effectue en exécutant l'instruction courante est en prenant comme nouvelle instruction courante celle qui est donnée par le **goto** associé.

1. Montrer qu'une machine à deux compteurs peut simuler une machine à une pile.
2. En déduire qu'une machine à quatre compteurs peut simuler une machine à deux piles.
3. Montrer qu'une machine à n compteurs peut être simulée par une machine à deux compteurs.

On peut démontrer que si une machine à deux compteurs contient n dans l'un de ces registres et 0 dans l'autre alors il n'est pas possible qu'elle s'arrête avec un registre contenant 2^n et l'autre 0. Expliquer pourquoi ce théorème n'est pas en contradiction avec les résultats obtenus plus haut.

Exercice 2.5

On appelle RAM simple une RAM qui n'a pas d'instruction read(n) ni d'instruction write(n). On dit que $f : \mathbb{N} \to \mathbb{N}$ est calculée par la RAM simple M ssi, lorque l'on lance M avec un contenu du registre R_0 égal à n, son calcule s'arrête si et seulement si $n \in dom(f)$ et dans ce cas le contenu de R_0 est $f(n)$.

1. Montrer qu'une machine RAM simple peut simuler une machine à k compteurs.
2. Montrer qu'une machine à k compteurs peut simuler une RAM simple.

Exercice 2.6

Étant donné un alphabet fini Σ, on appelle *système de réécriture sur* Σ un sous-ensemble fini de $\Sigma^+ \times \Sigma^*$. Si R est un système de réécriture sur Σ on dit que R réécrit la chaîne s_1 en s_2 (ce que l'on note $s_1 \to_R s_2$) si il existe (w_1, w_2) dans R tel que $s_1 = v_1 w_1 v_2$ et $s_2 = v_1 w_2 v_2$. On dit que la chaîne s_r est le résultat du calcul de R à partir de s_i si il existe une suite finie $(s_k)_{0 \leq k \leq n}$ telle que $s_0 = s_i$, $s_r = s_n$, pour tout $k < n$, $s_k \to_R s_{k+1}$ et il n'existe pas de t tel que $s_r \to_R t$.

1. Simuler une machine de Turing à l'aide d'un système de réécriture.
2. Simuler un système de réécriture à l'aide d'une machine de Turing non-déterministe.