TD 5 : Machines RAM et théorème de Rice

{lionel.rieg,paolo.tranquilli}@ens-lyon.fr

Exercice 1. *Machines RAM*

Une RAM [1] est une machine abstraite opérant sur un ensemble infini de registres notés R_i ($i \in \mathbb{N}$) contenant chacun un entier. Formellement, une RAM est définie par une suite finie de lignes contenant chacune une instruction et optionnellement étiquetées par un identificateur l. Ses instructions sont les suivantes :

1) $R_i \leftarrow 1$
2) $R_i \leftarrow R_j + R_k$
3) $R_i \leftarrow R_j \dot{-} R_k$
4) $R_{R_i} \leftarrow R_{R_j}$
5) `if` $R_i = 0$ `jmp` l
6) `stop`

où $n \dot{-} m$ est la soustraction bornée définie par $n \dot{-} m = \max(n-m, 0)$, et les indices i, j, k ne sont pas nécessairement distincts.

À chaque instant, la configuration d'une RAM est définie par le contenu de ses registres et le numéro de la ligne qui doit être exécutée. On suppose qu'au début de l'exécution tous les registres contiennent 0, sauf R_0 qui contient l'entrée. À l'exécution de l'instruction `stop` on considère le contenu de R_0 comme la sortie du programme.

1. Montrer qu'on peut programmer les instructions suivantes :

7) $R_i \leftarrow n$
8) $R_i \leftarrow R_j$
9) $R_i \leftarrow R_{R_j}$
10) $R_{R_i} \leftarrow R_j$
11) `jmp` l
12) `if` $R_i > 0$ `jmp` l
13) `if` $R_i = R_j$ `jmp` l
14) `if` $R_i < R_j$ `jmp` l
15) $R_i \leftarrow R_j \cdot R_k$
16) $R_i \leftarrow \lfloor R_j / n \rfloor$
17) $R_i \leftarrow R_j \bmod n$

2. Montrer que toute machine de Turing peut être simulée par une RAM, en supposant (ou codant) que l'alphabet Γ est $\{0, \ldots, |\Gamma| - 1\}$ et l'ensemble d'états Q est $\{0, \ldots, |Q| - 1\}$.

Nous allons montrer réciproquement que toute RAM peut être simulée par une machine de Turing. On en déduira que les fonctions calculables par ces deux modèles sont les mêmes.

3. Déterminer une traduction des instructions non conditionnelles vers les machines de Turing, en faisant attention à toujours utiliser le même nombre de rubans.

4. Donner une traduction d'une RAM en une machine de Turing.

[1] pour *Random Access Machine*

Exercice 2. *Théorème de Rice*

Théorème. *Soit P un sous-ensemble des fonctions (partielles) sur un alphabet Σ calculables par machine de Turing. Si P est non trivial, c.-à-d. non vide et non plein, alors déterminer si une machine de Turing M calcule une fonction de P est un problème indécidable.*

1. Est-ce que le théorème de Rice implique que les seules propriétés décidables sur des machines de Turing sont triviales ?

On va à présent montrer le théorème de Rice par réduction au problème de l'arrêt. On suppose donc qu'il existe une machine de Turing M_P qui décide P.

2. Justifier que l'on peut se restreindre au cas où P ne contient pas la fonction Θ nulle part définie.

3. Construire un programme dont l'appartenance à P (non trivial et ne contenant pas Θ) est équivalente à l'arrêt d'une machine de Turing M sur un mot donné w.

Exercice 3. *Applications du théorème de Rice*

Préciser si le théorème de Rice s'applique aux problèmes suivants :

1. Est-ce que la machine de Turing M s'arrête sur le mot vide ?

2. Est-ce que la machine de Turing M calcule la fonction carré ?

3. Est-ce que la machine de Turing M sur l'entrée n fixée calcule n^2 ?

4. Est-ce que la machine de Turing M s'arrête en moins de 1000 pas sur le mot vide ?

5. Est-ce que la machine de Turing M sur l'entrée n fixée calcule n^2 en moins de $n^2 + 1000$ pas de calcul ?

6. Est-ce que la machine de Turing M calcule n^2 en moins de $n^2 + 1000$ pas de calcul pour toute entrée n ?

7. Est-ce que la machine de Turing M a un indice $\langle M \rangle$ pair ?

8. Est-ce que deux machines de Turing M et N calculent la même fonction ?

9. Est-ce qu'une RAM va écrire dans un registre donné R_i pour quelque donnée en entrée ?