

Calculabilité et Complexité

TD 2 – Autres modèles de calcul

Exercice 1 : Le modèle RAM

Les machines RAM (*Random Access Machine*) représentent un autre modèle pour la calculabilité et la complexité. Ce modèle se rapproche dans son mode de fonctionnement des machines physiques en utilisant la notion de programme, de registres, de compteur ordinal et de jeu d'instruction réduit.

Les éléments constitutifs d'une machine RAM sont :

Deux bandes : l'une de lecture, servant pour les entrées, et l'autre d'écriture pour sortir le/les résultats ;

Les registres : ce sont des cases mémoire contenant des entiers positifs. Il existe trois genres de registres :

un compteur ordinal : il contient le numéro de l'instruction suivante à exécuter ;

un accumulateur : sur lequel on pourra agir ;

des registres *normaux* : ils sont indicés par N . Pour représenter le n -ème registre, on notera r_n .

Les instructions : elles permettent de manipuler les registres, d'effectuer des opérations arithmétiques, de réaliser des instructions de contrôle ou d'effectuer des entrées sorties.

Manipulation de registres : L'instruction **load** permet de stocker une valeur d'un registre vers l'accumulateur, **store** permet de stocker le contenu de l'accumulateur dans un registre selon plusieurs modes :

- **load #n** (adressage absolu) : met l'entier n dans l'accumulateur ;
- **load n** (resp. **store n**) (adressage direct) cela a pour effet de charger le contenu du registre r_n dans l'accumulateur (resp. d'enregistrer le contenu de l'accumulateur dans le registre r_n ;
- **load (n)** (resp., **store (n)**) (adressage indirect), si p est le contenu de r_n , alors cela charge le contenu de r_p dans l'accumulateur.

Les opérations arithmétiques :

- **incr** incrémente ce qu'il y a dans l'accumulateur
- **decr** décrémente ce qu'il y a dans l'accumulateur s'il est positif, et le laisse inchangé s'il est nul.

Rupture de séquences :

- **stop** provoque l'arrêt de la machine
- **jump** saut inconditionnel
- **jumpz** saut conditionnel si la valeur de l'accumulateur est 0

Entrées sorties :

- **read** lit un élément de la bande d'entrée
- **write** écrit un élément sur la bande de sortie

Le fonctionnement de la machine est simple. On exécute successivement les instructions du programme (sauf si on exécute un saut), on arrête le programme lorsque l'on rencontre une instruction **stop** ou que l'on arrive en fin de programme.

Écrire un programme qui calcule la somme de deux nombres entiers lus sur le ruban d'entrée.

Exercice 2 : Équivalence avec les machines de Turing

a : Montrer que l'exécution de tout programme RAM peut être simulée par une machine de Turing.

b : Montrer que tout programme d'une machine de Turing peut être simulé par un programme RAM. Pour cela, on s'intéressera aux points suivants :

1. choix de la MT
2. initialisation de la machine RAM
3. simulation du comportement de la machine de Turing

Exercice 3 : Machines de Turing non déterministes

Une machine de Turing non déterministe possède la même modélisation générale que les machines déterministes. La différence provient de la fonction de transition Δ qui est un sous-ensemble de $(Q \times \Sigma) \times (Q \times \Sigma \times \{G, St, D\})$.

Comme dans le cas des automates finis, un mot x de Σ^* est reconnu par une machine de Turing non déterministe s'il existe une suite finie de transitions qui se termine sur un état d'acceptation.

Il se peut que certaines *exécutions* de cette machine sur une entrée donnée ne termine pas alors que le mot est reconnu.

Le but de cet exercice est de montrer que tout langage accepté par une machine de Turing non déterministe l'est aussi par une machine de Turing déterministe.

a : Montrer qu'une suite d'entiers de longueur i inférieurs à r correspond à une exécution de la machine de Turing non déterministe de longueur i .

b : soit r le nombre maximal de transitions à partir d'un état q et d'une lettre a . Montrer que l'on peut énumérer successivement par une machine de Turing toutes les suites de longueur inférieure à i d'entiers inférieurs à r .

c : Définir une machine de Turing à 3 rubans qui permet de répondre à la question.