

TD3 : Modèles de calcul

MT sur un ruban semi-infini

Une MT sur ruban semi-infini est une machine de Turing qui fonctionne sur un ruban indexé par \mathbb{N} au lieu de \mathbb{Z} . On suppose que la règle de la MT n'amène jamais la tête à sortir du ruban.

Remarque : On ne suppose pas a priori que la tête de la MT est capable de détecter qu'elle se trouve en bord de ruban, donc le format de la règle est inchangé.

Exercice 1. Montrer que les MT calculent exactement les mêmes fonctions que les MT sur ruban semi-infini.

MT à k rubans

Une machine de Turing à k rubans \mathcal{M} est un tuple $(Q, \Gamma, \Sigma, \delta, q_0, B, q_F)$ similaire à celui d'une MT classique mais tel que :

- Chaque ruban possède une tête, donc la machine lit k symboles de Γ à chaque étape.
- La fonction de transition partielle est $\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{\leftarrow, \downarrow, \rightarrow\}^k$.
- Une configuration est un élément (q, u, z) de $Q \times (\Gamma^k)^{\mathbb{Z}} \times \mathbb{Z}^k$.
- La configuration initiale associée à l'entrée $w \in \Sigma^*$ est $(q_0, (\omega B \cdot w B^\omega, \omega B^\omega, \dots, \omega B^\omega), (0, 0, \dots, 0))$.
- Pour le calcul d'une fonction, la sortie est lue sur le premier ruban entre la position de la tête et le premier symbole B sur la droite.

Exercice 2. Montrer que les MT à 1 ruban calculent exactement les mêmes fonctions que les MT à k rubans.

Machine RAM

Une machine RAM est constituée :

- d'un programme, c'est à dire une liste finie d'instructions numérotées ;
- d'un registre appelé *co* contenant le compteur ordinal, c'est à dire le numéro de la prochaine instruction à exécuter, ce registre est incrémenté après chaque instruction (sauf en cas de saut) ;
- de k registres non bornés R_0, \dots, R_{k-1} , chacun contenant un entier positif ou nul ;
- d'une mémoire constituée d'une infinité de cases adressées à partir de 0, chacune pouvant contenir un entier.

L'entrée du calcul est placée dans le registre R_0 et la sortie est ce qui s'y trouve à la fin du calcul.

Les instructions du modèle sont regroupées de la manière suivante :

1. Pour déplacer des valeurs

- CONST n charge l'entier n dans R_0 ;
- LOAD n charge le contenu de la case d'adresse R_n de la mémoire dans R_0 ;
- STORE n charge le contenu de R_0 dans la case d'adresse R_n de la mémoire ;

- **MOVE n** charge le contenu de R_n dans R_0 ;
- **COPY n** charge le contenu de R_0 dans R_n ;

2. Opérations arithmétiques

- **ADD n** ajoute le contenu du registre R_n à R_0 ;
- **SUB n** soustrait le contenu du registre R_n à R_0 (en laissant 0 si le résultat est négatif).

3. Déplacements

- **JUMP n** charge l'entier n dans le registre co (saute à l'instruction n) ;
- **JUMPZ n** charge l'entier n dans le registre co si R_0 contient 0, rien sinon ;
- **HALT** fin du programme, la sortie est lue dans R_0 .

Exercice 3. Donner le programme d'une machine RAM qui, sur l'entrée n , calcule la valeur u_n de la suite définie par $u_{n+1} = u_n + u_{n/2}$ avec $u_0 = u_1 = 1$.

Exercice 4. Montrer que toute fonction calculée par une machine RAM est calculable par une MT (sans contrainte sur le nombre de rubans).

Remarque : On laisse la démonstration de la simulation inverse en entraînement.