# Maitrise et IUP d'informatique 1999-2000
# Université de Marne la Vallée
## Calculabilité. Devoir 1

*Le devoir est à rendre pour le 16 novembre. De préférence, on utilisera le langage de programmation des machines de Turing vu en cours. On mettra des commentaires.*

**Exercice 1.**— Étant donné un langage rationnel $L$ (et donc un automate fini déterministe qui le reconnaît), construire une machine de Turing qui reconnaît les mots de $L$.

**Exercice 2.**— Montrer, d'une façon différente de celle vue en cours, que l'on peut simuler une machine de Turing $M$ à une bande bi-infinie par une machine de Turing $M'$ à une bande infinie à droite. Pour repérer le début de la bande infinie à droite, on pourra utiliser un marqueur $ n'apparaissant pas ailleurs comme symbole. Les cases de la machine $M$ étant numérotées par des entiers relatifs $(..., -2, -1, 0, 1, 2, ...)$, $M'$ pourra utiliser des décalages vers la droite à chaque fois que $M$ accède à une nouvelle case négative.

**Exercice 3.**— On considère un nouveau modèle de machine de Turing à une bande : une machine de Turing de ce modèle peut non seulement écraser un symbole à la position de sa tête de lecture/écriture en le remplaçant par un autre, mais aussi supprimer le symbole sous cette tête de lecture/écriture ou insérer un nouveau symbole.
— Définir précisément la fonction de transition et le calcul de telles machines de Turing.
— Une telle machine de Turing peut-elle être simulée par une machine de Turing ordinaire ? Si oui, avec quelle perte de temps ?

**Exercice 4.**— On considère un modèle de machine de Turing à une bande où la tête de lecture/écriture se place non pas sur une case, mais entre deux cases de la bande. La machine peut consulter simultanément ces deux cases, et peut à chaque transition écraser les deux symboles lus par deux nouveaux symboles. Répondre aux mêmes questions que dans l'exercice précédent.

*Dans les exercices qui suivent, on demande de construire des machines de Turing non-déterministes.*

**Exercice 5.**— On a $p$ processeurs qui doivent exécuter en parallèle $q$ tâches differentes en un temps global maximal $t$. Les $q$ tâches sont exécutées en un temps $t_1, t_2, \ldots, t_q$ respectivement. Une fois qu'un processeur commence une tâche, il l'exécute complètement. Peut-on décider avec une machine de Turing non-déterministe si on peut répartir les $q$ tâches sur les $p$ processeurs de manière

à ce que l'exécution totale se fasse en temps $\leq t$.
Indication : la solution sera générée sur une des bandes de la machine sous la forme : $p_1\#p_2\#\ldots\#p_i\#\ldots\#p_q$, où $p_i$ représente le numéro du processus qui doit exécuter la tâche $i$.

**Exercice 6.**— On considère le langage de mots sur $\{a,b\}^*$ suivant :

$$L = \{u.ab^pa.v.ab^pa.w \mid u,v,w \in \{a,b\}^* \texttt{ et } p \geq 0\}$$

Décrire une machine de Turing non-déterministe qui décide $L$ en temps $n$ et en espace $\log(n)$. On rappelle que la complexité en temps est la longueur maximale d'un calcul acceptant dans la machine non-déterministe.

**Exercice 7.**— Le problème du mariage parfait est le suivant :
**Données :** Un entier $n$ et une matrice $n \times n$ à coefficients dans $\{0,1\}$.
**Question :** Peut-on remplacer un certain nombre de 1 par 0 dans la matrice donnée pour obtenir une matrice ayant exactement une entrée 1 par ligne et par colonne ?
— Expliquer comment coder une donnée du mariage parfait.
— Donner une machine de Turing non-déterministe pour résoudre le problème du mariage parfait. Quelles sont les complexités en temps et espace de cette machine de Turing ?