

# Projet Java, L3 Miage, 2011-2012 :

## La machine de Turing

### Contraintes :

- A rendre pour le 6 janvier 2011 (délai de rigueur)
- A réaliser par groupe de trois personnes maximum (envoyez moi un mail afin de me signaler le nom des membres du groupe [pechoux@loria.fr](mailto:pechoux@loria.fr)).

### Description du sujet :

Ce projet consiste à réaliser un petit programme permettant de construire et simuler une machine de Turing à un seul ruban. Il fait appel aux connaissances qui vous ont été dispensées lors de vos cours de programmation objet, d'outils conceptuels et à votre capacité à vous documenter en utilisant les API Java([java.sun.com](http://java.sun.com)).

Formellement, une machine de Turing se compose d'un ensemble d'états  $K$  incluant un état initial  $s$ , d'un alphabet  $\Sigma$  et d'une fonction de transition  $\delta$  qui, à un état dans  $K$  et une lettre de l'alphabet dans  $\Sigma$ , associe un état dans  $K$ , une lettre dans  $\Sigma$  et un mouvement dans l'ensemble  $\{G, D, I\}$  (N.B. pour gauche, droite, immobile).

La machine de Turing possède une tête de lecture qui peut parcourir les symboles du ruban. Si  $q$  est l'État courant et  $s$  est le symbole lu par la tête de lecture alors :

$$\delta(q,s)=(p,r,D)$$

signifie que  $p$  est le nouvel État, que  $r$  est le symbole qui remplace  $s$  sur le ruban et que la tête de lecture se déplace de  $D$  (dans  $\{G, D, I\}$ ).

L'alphabet sera restreint à  $\{0,1,e\}$ ,  $e$  représente un nouvel élément lorsque l'on déplace la tête de lecture au delà des limites d'un mot (parfois appelé  $B$  ou blank).

### Consignes (étapes) :

1. Choisir une modélisation adéquate pour la machine de Turing incluant le ruban, les états, la table de transition.
2. Ecrire au moins un constructeur, une méthode `toString` pour chacune des classes, une méthode `step` permettant de simuler une étape de la machine, une méthode `execute` permettant de simuler la machine jusqu'à son arrêt éventuel.
3. Inclure une méthode affichant l'évolution de la machine (valeur du ruban, État courant, position de la tête) pour chaque transition
4. Définissez un certain nombre de constantes (par exemple, l'Élément  $e$ ) dans une interface que votre programme implémentera.
5. Ecrire une machine de Turing calculant la division d'un nombre binaire par 2.
6. Ecrire une machine de Turing calculant l'addition de deux nombres binaires.
7. Ecrire une machine de Turing calculant la multiplication de deux nombres binaires.
8. Comparer le temps d'exécution de chacune de ces machines sur des entrées de taille  $n$  (la taille étant la longueur de l'entrée).

Vous devrez rendre :

- une archive .zip contenant votre programme (.class et .java),
- un fichier exécutable comprenant des données afin de tester votre programme.
- et un rapport au format pdf (de 4 pages -maximum strict !-, page de garde exclue) expliquant vos choix, les difficultés rencontrées (une javadoc n'est pas demandée).

**Evaluation :**

L'évaluation portera sur :

1. Le respect des consignes (3pt)
2. La clarté, l'optimalité du code et la correction du code (7pt)
3. L'utilisation des concepts orientés objet étudiés en cours (héritage, polymorphisme, classes abstraites, interfaces, exceptions...). (3pt)
4. La clarté et la pertinence du rapport (7pt)