

M1 Informatique – Université du Havre

# Informatique Théorique

Machine de Turing

Florian Aline  
31/01/2016

## Introduction

Une machine de Turing se compose d'une partie de contrôle et d'une bande infinie sur laquelle se trouvent écrits des symboles. La partie de contrôle est constituée d'un nombre fini d'états possibles et de transitions qui régissent les calculs de la machine. Les symboles de la bande sont lus et écrits par l'intermédiaire d'une tête de lecture.

La bande joue à la fois le rôle de périphérique d'entrée et de périphérique de sortie. Initialement vierge, on y inscrit les données à traiter par le programme, qui sont des mots sur un alphabet  $\Sigma$ . Le premier symbole du premier mot est placé dans la cellule d'indice 0 où se trouve la tête de lecture/écriture. Pendant l'exécution du programme, la tête de lecture/écriture peut se déplacer à gauche ou à droite et à la fin de l'exécution du programme, les mots sur la bande constituent le résultat du calcul.

## Travail réalisé

Le travail demandé est de d'écrire un programme qui prend en entrée "une machine de Turing accompagnée de sa donnée initiale", supposée fonctionner en mode calculateur, et simule l'exécution de cette dernière étape par étape jusqu'au résultat final.

Pour faire cela, j'ai dû créer trois classes.

- *Transition.java* : Dans cette classe, on s'occupe de gérer les différentes transitions de la bande pour passer d'un état à un autre et de ce fait, d'un symbole à un autre. Il nous faut donc le nombre requis pour passer d'un état à un autre, le nœud qui suivra notre transition, la valeur de ce dernier et ainsi que le mouvement que va effectuer notre tête de lecture.
- *Nœud.java* : Dans cette classe, on stock simplement toutes les transitions possibles lorsque nous sommes dans un état donné. Ainsi, si un état admet plusieurs symboles pour passer dans un autre état, ces transitions seront stockées dans cette classe Nœud. Chaque état à son nœud et un simple attribut listeTransitions permet de stocker toutes les données que l'on veut.
- *Turing.java* : Cette classe s'occupe du calcul de la machine de Turing qui se forme grâce aux paramètres fournis. On initialise alors, notre bande, nos différents nœuds, puis on initialise nos transitions. Pour faire cela, on traite les données en binaire passées en paramètre pour les décoder et créer alors nos transitions. Une fois cela fait, on lance alors notre calcul. Tant que notre machine peut continuer à calculer, on effectue le traitement en regardant l'état dans lequel on est et ces différentes transitions pour voir si cela correspond à ce qu'on lit sur la bande. A chaque étape, la bande est affichée pour montrer les différents traitements réalisés dessus.

## Code source du programme.

```
/**
 * Aline Florian
 * Informatique théorique
 * Machine de Turing
 * M1 Informatique
 * Université du Havre
 */

public class Transition {

    private int mouvement;
    private int nombre;
    private int noeudSuiv;
    private int valeur;

    public Transition(int nombre, int noeudSuiv, int valeur, int mouvement) {
        this.nombre = nombre;
        this.noeudSuiv = noeudSuiv;
        this.valeur = valeur;
        this.mouvement = mouvement;
    }

    public int getMouvement() {
        return mouvement;
    }

    public void setMouvement(int mouvement) {
        this.mouvement = mouvement;
    }

    public int getNombre() {
        return nombre;
    }

    public void setNombre(int nombre) {
        this.nombre = nombre;
    }

    public int getNoeudSuiv() {
        return noeudSuiv;
    }

    public void setNoeudSuiv(int noeudSuiv) {
        this.noeudSuiv = noeudSuiv;
    }

    public int getValeur() {
        return valeur;
    }

    public void setValeur(int valeur) {
        this.valeur = valeur;
    }
}
```

```
import java.util.ArrayList;
import java.util.List;
```

```
/**
 * Alline Florian
 * Informatique théorique
 * Machine de Turing
 * M1 Informatique
 * Université du Havre
 */
```

```
public class Noeud {
    private ArrayList<Transition> listeTransitions;

    public Noeud(){
        listeTransitions = new ArrayList<Transition>();
    }

    public List<Transition> getTransitions(){
        return listeTransitions;
    }

    public void addTransition(int nombre, int noeudSuivant, int valeur, int mouvement){

        Transition t = new Transition(nombre,noeudSuivant,valeur, mouvement);
        listeTransitions.add(t);

    }
}
```

```
import java.util.ArrayList;
import java.util.regex.Pattern;
```

```
/**
 * Alline Florian
 * Informatique théorique
 * Machine de Turing
 * M1 Informatique
 * Université du Havre
 */
```

```
public class Turing {

    private int index;
    private int positionNoeud;

    private String etatInitial;

    private ArrayList<String> bande;
    private Noeud[] noeuds;

    public Turing(String transitions,String etatInitial,String bande){

        this.etatInitial=etatInitial;
        this.bande=new ArrayList<String>();
        this.bande.add(bande);
        this.noeuds = new Noeud[5];
    }
}
```

```

initialiserTransitions(transitions);
positionNoeud = Integer.parseInt(etatInitial,2);

for(int i=0;i<200;i++)
    this.bande.add("3");

for(int i=0;i<bande.length();i++)
    this.bande.set(i,""+bande.charAt(i));
}

private void initialiserTransitions(String transitions){

    String[] tabTransi = transitions.split(Pattern.quote(",("));
    Integer noeudCourant,nombre,noeudSuivant,valeur,mouvement;
    String[] str;

    for(int i=0;i<tabTransi.length;i++){
        tabTransi[i] = tabTransi[i].replace("(", "").replace(")", "");
    }

    for(String s : tabTransi){

        str = s.split(Pattern.quote(",("));

        noeudCourant = Integer.parseInt(str[0],2);
        noeudSuivant = Integer.parseInt(str[2],2);
        mouvement = Integer.parseInt(str[4],2);

        if(str[1].equals("3"))
            nombre = 3;
        else
            nombre = Integer.parseInt(str[1],2);

        if(str[3].equals("3"))
            valeur = 3;
        else
            valeur = Integer.parseInt(str[3], 2);

        if(noeudCourant>noeuds.length || noeuds[noeudCourant] == null){

            noeuds[noeudCourant] = new Noeud();
            noeuds[noeudCourant].addTransition(nombre,noeudSuivant,valeur,mouvement);
        }
        else
            noeuds[noeudCourant].addTransition(nombre,noeudSuivant,valeur,mouvement);
    }
}

```

```

public ArrayList<String> calculer(){

    int lecture;

    boolean estBloque=false;
    boolean verif=false;

    while(!estBloque){

        if(index<0)
            lecture=3;
        else
            if(bande.get(index).equals("3"))
                lecture=3;
            else
                lecture = Integer.parseInt(bande.get(index),2);

        boucleInterne:
        for(Transition transi : noeuds[positionNoeud].getTransitions()){

            if(transi.getNombre() == lecture){

                if(index== -1)
                    bande.add(0,""+transi.getValeur());
                else
                    bande.set(index,""+transi.getValeur());

                if(transi.getMouvement()==0)
                    index--;
                else
                    index++;

                positionNoeud=transi.getNoeudSuiv();
                verif=true;

                break boucleInterne;
            }
        }

        estBloque= !verif;
        verif=false;

        System.out.print("Bande : \n");

        afficherBande(index);
    }

    System.out.print("Bande : \n");

    afficherBande(index);

    return this.bande;
}

```

```

private void afficherBande(int index){

    int cpt=0;

    System.out.println("-----");

    for(String s : this.bande){

        if(s.equals("3")){
            cpt++;
            System.out.print("# ");
        }
        else{
            cpt=0;
            System.out.print(s + " ");
        }
        if(cpt==2) break;

    }

    System.out.println("\nPosition de la tête : " + index);
    System.out.println("-----");
    System.out.println();
}

public static void main(String[] args) {
    String transitions =
"(0,0,0,1,1),(0,3,1,3,0),(1,1,10,0,0),(10,1,10,1,0),(10,0,10,0,0),(10,3,11,0,1),(11,0,11,0,1),(11,1,10,0,0)";
    String etatInitial = "0";
    String bande = "00";
    Turing machineTuring = new Turing(transitions,etatInitial,bande);

    System.out.println("Debut de l'execution\n");
    System.out.println("Bande Initiale :");
    machineTuring.afficherBande(0);
    machineTuring.calculer();
    System.out.println("Fin de l'execution");
}
}

```

## Trace d'exécution

Trace n°1

Bande initiale : 00

Objectif : Doubler les 0 de la bande (pas de 1)

Debut de l'execution

Bande Initiale :

-----  
0 0 # #  
Position de la tête : 0  
-----

Bande :

-----  
1 0 # #  
Position de la tête : 1  
-----

Bande :

-----  
1 1 # #  
Position de la tête : 2  
-----

Bande :

-----  
1 1 # #  
Position de la tête : 1  
-----

Bande :

-----  
1 0 # #  
Position de la tête : 0  
-----

Bande :

-----  
1 0 # #  
Position de la tête : -1  
-----

Bande :

-----  
0 1 0 # #  
Position de la tête : 0  
-----

Bande :

-----  
0 1 0 # #  
Position de la tête : 1  
-----

Bande :

-----  
0 0 0 # #  
Position de la tête : 0  
-----



Bande :

-----  
0 0 0 # #  
Position de la tête : -1  
-----

Bande :

-----  
0 0 0 0 # #  
Position de la tête : 0  
-----

Bande :

-----  
0 0 0 0 # #  
Position de la tête : 1  
-----

Bande :

-----  
0 0 0 0 # #  
Position de la tête : 2  
-----

Bande :

-----  
0 0 0 0 # #  
Position de la tête : 3  
-----

Bande :

-----  
0 0 0 0 # #  
Position de la tête : 4  
-----

Bande :

-----  
0 0 0 0 # #  
Position de la tête : 4  
-----

Bande :

-----  
0 0 0 0 # #  
Position de la tête : 4  
-----

Fin de l'exécution

## Trace n°2

Bande initiale : 01

Objectif : Doubler les 0 de la bande (pas de 1)

Final : La machine se bloque à cause du 1 et elle s'arrête.

Debut de l'execution

Bande Initiale :

```
-----  
0 1 # #  
Position de la tête : 0  
-----
```

Bande :

```
-----  
1 1 # #  
Position de la tête : 1  
-----
```

Bande :

```
-----  
1 1 # #  
Position de la tête : 1  
-----
```

Bande :

```
-----  
1 1 # #  
Position de la tête : 1  
-----
```

Fin de l'execution