

Implémentation des automates

F. Barthélemy

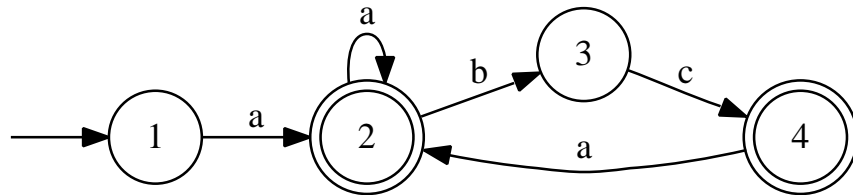
19 octobre 2010

Introduction

On va se donner comme problème : écrire un programme qui teste si une chaîne donnée appartient au langage d'un automate fini déterministe donné. Cette chaîne sera donnée sous forme d'une chaîne de caractères.

1 Automate codé dans les instructions du programme

C'est une démarche qu'on peut suivre lorsqu'on a un petit automate immuable. Les transitions sont transcrites par des branchements conditionnels.



Exemple :

```
public class EncodeAutoCode{
    static boolean automate(String s){
        int etat = 1;
        for (int i=0; i<s.length(); i++){
            System.out.println("——" + etat + "␣" + s.charAt(i));
            switch(s.charAt(i)){
                case 'a':
                    if (etat==1 || etat==2 || etat==4){
                        etat = 2;
                        break;
                    } else
                        return false;
                case 'b':
                    if (etat==2){
                        etat=3;
                    }
            }
        }
    }
}
```

```

        break;
    }
    else
        return false;
case 'c':
    if (etat==3){
        etat = 4;
        break;
    }else
        return false;
default:
    System.out.println("Caractère hors alphabet : " + s.charAt(i));
    return false;
}
}
System.out.println("—" + etat);
return (etat==2)|| (etat==4);
}
public static void main(String[] args){
    boolean res;
    if (args.length==0)
        System.out.println("chaînes à tester attendues sur la ligne de commande");
    for (int i = 0; i < args.length; i++){
        System.out.println("Chaîne testée : " + args[i]);
        res = automate(args[i]);
        System.out.println("Résultat : " + res);
    }
}
}

```

Exemple d'exécution :

```

> java EncodeAutoCode aaa aab aabc aaabcabcabc adz
Chaîne testée: aaa
---1 a
---2 a
---2 a
---2
Résultat: true
Chaîne testée: aab
---1 a
---2 a
---2 b
---3
Résultat: false

```

```

Chaîne testée: aabc
---1 a
---2 a
---2 b
---3 c
---4
Résultat: true
Chaîne testée: aaabcabcabc
---1 a
---2 a
---2 a
---2 b
---3 c
---4 a
---2 b
---3 c
---4 a
---2 b
---3 c
---4
Résultat: true
Chaîne testée: adz
---1 a
---2 d
Caractère hors alphabet: d
Résultat: false

```

2 Implementation d'un automate comme une donnée

La représentation de l'automate n'est pas dans les instructions mais dans des variables. Il faut alors un programme pour interpréter ces données et produire une exécution. Ce programme est le même pour tous les automates codé de la même façon.

Une représentation classique : au moyen de tableaux. Un tableau à deux dimensions pour les transitions et un tableau de booléens pour le caractère final ou non des états.

Pour le même automate qu'à la section précédente, les deux tableaux en question sont :

état de départ	a	b	c
1	2		
2	2	3	
3			4
4	2		

et

etat	1	2	3	4
final	false	true	false	true

Pour une représentation en java, il faut adapter la représentation parce que les tableaux sont indicés par des entiers à partir de 0. Il faut un encodage des caractères de l'alphabet et par ailleurs retrancher 1 au numéro de l'état. De plus, on va remplir les cases vides avec -1.

```

public class EncodeAutoSDD{
    static int charToInt(char c){
        switch(c){
            case 'a': return 0;
            case 'b': return 1;
            case 'c': return 2;
            default: return -1;
        }
    }
    static boolean automate(String s, int [][] tr, boolean [] ef){
        int etat = 0;
        for (int i=0; i<s.length(); i++){
            System.out.println("——" + (etat+1) + "  " + s.charAt(i));
            int lechar = charToInt(s.charAt(i));
            if (lechar == -1){
                System.out.println("Caractère hors alphabet : " + s.charAt(i));
                return false;
            }
            etat = tr[etat][charToInt(s.charAt(i))];
            if (etat==-1)
                return false;
        }
        System.out.println("——" + (etat+1));
        return ef[etat];
    }
    public static void main(String [] args){
        int [][] transitions = {{1,-1,-1},{1,2,-1},{-1,-1,3},{1,-1,-1}};
        boolean [] etatfinal = {false, true, false, true};
        boolean res;
        if (args.length==0)
            System.out.println("chaînes à tester attendues sur la ligne de commande");
        for (int i = 0; i < args.length; i++){
            System.out.println("Chaîne testée : " + args[i]);
            res = automate(args[i], transitions, etatfinal);
            System.out.println("Résultat : "+res);
        }
    }
}

```

Et voici un exemple d'exécution.

```
> java EncodeAutoSDD aabc aaabcabcabc adz
Chaîne testée: aabc
---1 a
---2 a
---2 b
---3 c
---4
Résultat: true
Chaîne testée: aaabcabcabc
---1 a
---2 a
---2 a
---2 b
---3 c
---4 a
---2 b
---3 c
---4 a
---2 b
---3 c
---4
Résultat: true
Chaîne testée: adz
---1 a
---2 d
Caractère hors alphabet: d
Résultat: false
```