

Enoncé du TP1 (2021-2022)

Automates

On se propose de coder en langage C un automate fini déterministe complet $M = (Q, \Sigma, \delta, q_0, F)$, sur un alphabet binaire $\Sigma = \{0,1\}$. Q

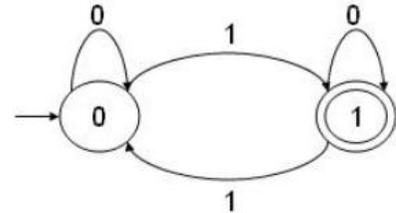
- Soit $Q = \{0, 1, \dots, n - 1\}$, $n = \text{card}(Q)$, le nombre des états de M
- Une transition sera codée par une structure composée de deux entiers (l'état de départ et l'état d'arrivée) et d'un caractère (le symbole d'entrée, qui est soit 0, soit 1).
- La fonction de transition δ sera codée par un tableau (les $2n$ transitions) qui est un pointeur vers la structure représentant une transition.
- L'état initial sera représenté conventionnellement par l'entier « 0 », ($q_0 = 0$).
- L'ensemble des états finaux sera codée par un entier (le nombre des états finaux) et un tableau (la liste des états finaux).

Les informations sur un automate seront stockées dans un fichier texte ayant le format suivant :

- Première ligne, le nombre des états.
- Les lignes suivantes contiennent chacune la description d'une transition sous forme d'un triplet : état de départ, symbole d'entrée, puis état d'arrivée.
- La dernière ligne contient, le nombre des états finaux, suivi par la suite des états finaux.

Par exemple, l'automate de la figure en face, sera stocké dans un fichier texte dont le contenu est :

```
2
0 0 0
0 1 1
1 0 1
1 1 0
1 1
```



Questions

1. Définir les deux types : transition et automate.
2. Ecrire les fonctions suivantes
 - a. `automate lire(const char* nom_fichier)` qui lit un automate depuis un fichier texte et renvoie sa structure.
 - b. `void afficher(const automate* M)` qui affiche les informations sur un automate donné M . La fonction de transition δ sera affichée comme une table.
 - c. `int delta_etoile(const char* w, const automate* M)` qui calcule la transition itérée δ^* du mot w à partir de l'état initial $q_0 = 0$ de l'automate M .
 - d. `int reconnu(const char* w, const automate* M)` qui test si le mot w est reconnu par un automate M .
 - e. `int vide(const automate* M)` qui test si le langage reconnu par un automate M est vide.
 - f. `int fini(const automate* M)` qui test si le langage reconnu par un automate M est fini.
 - g. `int infini(const automate* M)` qui test si le langage reconnu par un automate M est infini.