TP 2

Présentation

Le but de ce TP est d'approfondir l'utilisation des types structurés et des matrices, via l'étude d'automates finis.

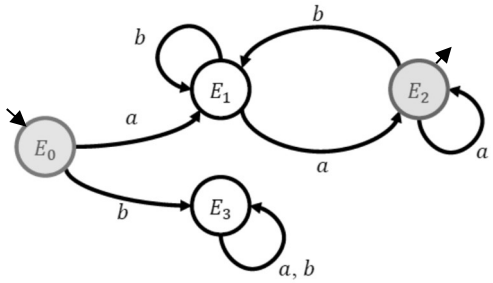
Définition

Un automate fini est un modèle mathématique d'ordinateur très simple, capable de reconnaître des *mots* formés de *lettres* contenus dans un *alphabet* prédéfini et respectant une certaine *structure*.

On peut représenter un automate par un *graphe orienté* :

- Chaque ***nœud*** s'appelle un *état* ;
- Chaque ***lien*** est un *changement d'état*, associé à une *lettre* de l'alphabet.

Exemple : soit l'automate représenté par le graphe suivant :



Cet automate comporte 4 états E_0, E_1, E_2 et E_3 :

- L'état E_0 représenté un état *initial*.
- L'état E_2 représenté un état *acceptant* (*Finaux*)
- Les états E_0, E_1 et E_3 sont des états *refusant*.

Les liens de cet automate sont associés aux lettres ***a*** ou ***b***, ce qui signifie qu'il est défini pour l'alphabet {***a***, ***b***}.

On peut représenter un automate sous la forme d'une *matrice de transitions*, qui indique les changements possibles pour chaque lettre, à partir de chaque état.

Exemple : pour l'automate précédent, on a la matrice de transition suivante :

	a	***b***
E_0	E_1	E_3
E_1	E_2	E_1
E_2	E_2	E_1
E_3	E_3	E_3

Si l'automate passé en argument est celui de l'exemple précédent, on obtiendra *exactement* l'affichage ci-dessous :

```
== Automate fini déterministe ==
Le nombre d'etats de l'automate est 4
Etats : E0 E1 E2 E3
Etat initial : E0
Etats Finauxs : E2
La matrice de transition est :
   a  b
E0 E1 E3
E1 E2 E1
E2 E2 E1
E3 E3 E3
```

NB. Dans ce TP, on veut implémenter un automate de ce type. Pour simplifier le problème, on fera les hypothèses suivantes :

- L'automate ne contient qu'un seul état initial et plusieurs états finaux;
- L'alphabet est toujours fini par exemple A={*a*, *b* } ; ou A ={*a*, *b* , c}

Implémentation

Écrivez la fonction `void saisis_automate(t_automate* a)` qui demande à l'utilisateur de saisir les différents champs constituant un automate. Les valeurs saisies sont utilisées pour initialiser l'automate passé en paramètre.

Exemple :

```
Entrez le nombre d'états : 4
Entrez les états (séparés par des espaces) : E0 E1 E2 E3
Entrez le nombre de symboles dans l'alphabet : 2
Entrez les symboles de l'alphabet (séparés par des espaces) : a b
Entrez le nombre de transitions : 8
Entrez les transitions (format : état symbole état_suivant) :
E0 a E1
E0 b E3
E1 a E2
E1 b E1
E2 a E2
E2 b E1
E3 a E3
E3 b E3
Entrez l'état initial : E0
Entrez le nombre d'états finaux : 1
Entrez les états finaux (séparés par des espaces) : E0
```

Tester votre programme avec l'automate suivant : E_0 état initiale et E_1, E_4 des états finaux (*acceptants*)