Déterminisation automatisé d'un AFN

Déroulement

Initialisation

Depuis le module automate `init_aut`
Plusieurs choix : 1. **Par défaut** 2. **Depuis un fichier**

Affichage de l'afn

Présente l'automate devant être déterminiser avec `aff_aut` dans le module afficheur qui utilise `aff_ens` afin d'afficher les éléments depuis une structure **ensemble** défini dans le module ensemble

Determinisation

Dans le module automate `det_aut` copie l'automate en effectuant les changements afin d'obtenir un automate deterministe, pour cela on utilise également `res_trans_d` qui résout le delta des transitions de l'automate déterministe et `tr_finaux` qui inscrit en tant qu'ensemble final ceux contenant un état final de l'afn.

Référencement des ensembles

Dans l'automate déterministe les transitions peuvent se diriger plusieurs fois vers un même ensemble d'états, c'est pourquoi ne seront noté que les indices pointant vers l'ensemble voulu à la place de ceux-ci.

Affichage de l'afd

ensemble

```
typedef struct {
  int ens[50];
}
```

Automate

```
typedef struct{
  ensemble q;
  char a[50];
  int t[50];
  ensemble i;
  ensemble f;
}Automate;
```

Automate_d

```
typedef struct{
  ensemble qd[50];
  char a[50];
  ensemble td[50];
  ensemble id;
  ensemble fd[50];
}Automate_d;
```

référencement d'ensemble
transitions
![transitions.png]

Éléments du programme

Module ensemble

Fonctions

eg_ens

```
int eg_ens(ensemble ens1,ensemble ens2)
```

- 1 si ens1 = ens2

est_ens

```
int est_ens(ensemble q[],ensemble e)
```

- 1 si e appartient à la liste d'ensemble q

est_etat

```
int est_etat(ensemble e,int n)
```

- 1 si l'état n fait partie de l'ensemble e

aj_ens

```
int aj_ens(ensemble q[],ensemble e)
```

- 1 si l'ensemble est ajouté la liste

aj_etat

```
int aj_etat(ensemble *e,int n)
```

- 1 si l'état est ajouté à l'ensemble

supp_etat

```
int supp_etat(ensemble *e,int n)
```

- 1 si un état supprimé
- 0 sinon **(l'état pouvait être absent)**

sep_ens_init

```
int sep_ens_init(ensemble i, int r[])
```

- Insère dans r les états initiaux convertis sous forme d'entier

trans

```
int trans(int t[],
          int ens_dep,
          char c,
          int ens_arr[])
```

- 1 si un état est renvoyé **(ens_arr)**
- Modifie tableau ens_arr et insère etat d'arrivée des transitions depuis etat_dep avec étiquette c
- Insère seulement à la suite de la table afin d'être conforme aux chemins d'un AFN

est_trans

```
int est_trans(int t[],
              int etat_dep,
              char etiq,
              int etat_arr)
```

- 1 si la transition existe

aj_trans

```
int aj_trans(int *t,
             int etat_dep,
             char etiq,
             int etat_arr)
```

- 1 si la transition a pu être ajoutée

trans_d

```
ensemble trans_d(Automate_d A,
                 ensemble ens_dep,
                 char etiq)
```

est_trans_d

```
int est_trans_d(ensemble td[],
```

```
                 ensemble ens_dep,
                 char etiq,
                 ensemble ens_arr)
```

aj_trans_d

```
int aj_trans_d(ensemble *td[],
               ensemble ens_dep,
               char etiq,
               ensemble ens_arr)
```

Module automate

Fonctions

A_defaut

```
void A_defaut(Automate *A)
```

- Insère l'automate de test par défaut dans A

A_fichier

```
int A_fichier(Automate *A)
```

- Insère l'automate configuré dans le fichier loader dans A

A_saisie

```
int A_saisie(Automate *A)
```

- Demande information par information les données à insérer dans l'automate A

init_aut

```
int init_aut(Automate *A)
```

- Propose à l'utilisateur le choix de la configuration de l'automate

1. A_defaut **automate par défaut**
2. A_fichier **automate lu dans le fichier *loader***
3. A_saisie **si l'envie vous prend de saisir l'automate en répondant aux questions**

res_trans_d

tr_finaux

det_aut

```
int det_aut(Automate A,Automate *B)
```

- Modifie l'automate B pour qu'il soit l'automate A determinisé

rec_mot

```
int rec_mot(Automate A,char mot[])
```

- 1 si le mot est reconnu par l'automate

Prog afficheur

à changer pour en faire un module
et faire un menu de choix pour manipuler l'automate

aff_ens

```
void aff_ens(ensemble e)
```

- Affiche un ensemble d'états e

aff_trans

```
int aff_trans(int t[])
```

- Affiche une liste de transitions d'un AFN

aff_aut

```
int aff_aut(Automate A)
```

- Affiche un AFN

aff_trans_d

```
int aff_trans_d(ensemble td[])
```

- Affiche un tableau de transition entre ensembles d'un AFD

aff_aut_d

```
int aff_aut_d(Automate_d a)
```

- Affiche un AFD