STLIB3

Generated by Doxygen 1.8.15

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 lts Struct Reference

Structure LTS.

`#include <automate.h>`

Public Attributes

- char ** **alphabet**
- int * etat

  tab ex manger, boire....// int eata[]
- int **etat_initial**
- transition * **tete**
- transition * **queue**
- int * etat_final

  tabde trans ex/t1, t2,...
- int nb_etat_final

  tab : ex 2, 4..
- int **nb_trans**

3.1.1 Detailed Description

Structure LTS.

Principale structure de notre programme , elle nous permet d'acceder a l'ensemble des donnees du programme.

The documentation for this struct was generated from the following file:

- automate.h

3.2 transition Struct Reference

Structure trnasiiton.

`#include <automate.h>`

Public Attributes

- int **depart**
- int **arrive**
- char * **alphabet**
- struct transition * **next**

3.2.1 Detailed Description

Structure trnasiiton.

Cette structure nous permettra d'acceder a un etat de depart , d'arrivee , aux alphabets , ainsi qu'a la transition suivante

The documentation for this struct was generated from the following file:

- automate.h

Chapter 4

File Documentation

4.1 automate.c File Reference

Implementation de l'ensemble de nos fonctions pour les LTS.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "automate.h"
```

Macros

- #define **MAX_T** 15

Functions

- void open_lts (char *filename)
  ouvre un fichier texte contenant un lts
- lts * read_lts (char *filename)
  lit les donnees d'un fichier lts
- lts * write_lts (char *filename)
  ecris un mot dans un fichier contenant un lts
- void sauver_lts (lts *l, const char *save)
  sauve un lts dans un fichier vide
- lts * creer_lts ()
  cree un lts vide
- int estVide (lts *l)
  verifie si un lts est vide
- int estpleine (lts *l)
  verifie si un lts est pleine
- void ajouter_etat (lts *l, int e)
  ajoute un etat e dans un lts
- void afficher_lts (lts *l)
  affiche un lts non-vide
- void ajouter_transition (lts *l, transition *t)
  ajoute une transition t dans un lts
- transition * new_transition (lts *l, char *al)
  creer une nouvelle transition , c'est comme si elle alloue de la memoire pour une transition donne .
- void **supprimer_transition** ()

4.1.1 Detailed Description

Implementation de l'ensemble de nos fonctions pour les LTS.

Author

Yoro , Awa , Firdaws , Kamilia

Version

1.@

Date

05 Janvier 2019

Cette fichier implemente les differentes fonctions pour la mise en place d'un lts de la creation a l'affichage en passant par l'ajout et la suppression des etats et transitions.

Author

Yoro , Awa , Firdaws , Kamilia

Version

1.@

Date

05 Janvier 2019

C'est dans ce fichier qu'on defini habituellement le prototype des fonctions qui seront implementes dans le fichier.c

4.1.2 Function Documentation

4.1.2.1 afficher_lts()

```
void afficher_lts (
            lts * l )
```

affiche un lts non-vide

Parameters

l	pointeur de type lts

4.1.2.2 ajouter_etat()

```
void ajouter_etat (
            lts * l,
            int e )
```

ajoute un etat e dans un lts

Parameters

l	pointeur de type lts
e	l'etat a ajouter de type int

4.1.2.3 ajouter_transition()

```
void ajouter_transition (
            lts * l,
            transition * t )
```

ajoute une transition t dans un lts

Parameters

l	pointeur de type lts
t	pointeur de type transition

4.1.2.4 estpleine()

```
int estpleine (
            lts * l )
```

verifie si un lts est pleine

Parameters

l	pointeur de type lts

Returns

Retourne 0 si le lts estpleine et 1 sinon

4.1.2.5 estVide()

```
int estVide (
            lts * l )
```

verifie si un lts est vide

Parameters

l	pointeur de type lts

Returns

Retourne 0 si le lts estvide et 1 sinon

4.1.2.6 new_transition()

```
transition* new_transition (
            lts * l,
            char * al )
```

creer une nouvelle transition , c'est comme si elle alloue de la memoire pour une transition donne .

Parameters

l	pointeur de type lts
al	pointeur de type char

4.1.2.7 open_lts()

```
void open_lts (
            char * filename )
```

ouvre un fichier texte contenant un lts

Parameters

filename	pointeur de type char

4.1.2.8 read_lts()

```
lts* read_lts (
            char * filename )
```

lit les donnees d'un fichier lts

Parameters

filename	pointeur de type char

Returns

Retourne un lts avec l'ensemble de ces donnees

4.1.2.9 sauver_lts()

```
void sauver_lts (
            lts * l,
            const char * save )
```

sauve un lts dans un fichier vide

Parameters

l	pointeur de type lts
save	pointeur de type char

4.1.2.10 write_lts()

```
lts* write_lts (
            char * filename )
```

ecris un mot dans un fichier contenant un lts

Parameters

filename	un fichier de type char pointeur

4.2 main.c File Reference

Fichier de test des fonctions du programme.

```
#include <stdio.h>
#include <stdlib.h>
#include "automate.c"
#include <string.h>
```

Functions

- int main ()

4.2.1 Detailed Description

Fichier de test des fonctions du programme.

Author

Yoro , Awa , Firdaws , Kamilia

Version

1.3

Date

05 Janvier 2019

Cette fichier permet de tester certains fonctions implments dans le fichier automate.c

4.2.2 Function Documentation

4.2.2.1 main()

```
int main ( )
```

supprimer_etat(l, t.depart);