

Doc Projet TL

Generated by Doxygen 1.8.6

Sun Feb 8 2015 20:53:06

Contents

1	Documentation du logiciel Autoroute	1
1.1	Introduction	1
1.2	Installation du logiciel pour les utilisateurs	1
1.3	Pour les développeurs	1
1.3.1	Etape 1 : Prise en main des sources et execution	1
1.3.2	Etape 2 : Arborescence du projet	2
1.4	Définitions	2
1.4.1	Minimisation d'un automate	2
1.4.2	Standardisation d'un automate	2
1.4.3	Produit de deux automates	2
1.4.4	Détermination de deux automates	2
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Class Documentation	9
5.1	Automate Class Reference	9
5.1.1	Constructor & Destructor Documentation	10
5.1.1.1	Automate	10
5.1.1.2	Automate	10
5.1.1.3	~Automate	10
5.1.2	Member Function Documentation	10
5.1.2.1	ajoutEtat	10
5.1.2.2	ajoutTransition	10
5.1.2.3	cible_transition	11
5.1.2.4	determinise	11
5.1.2.5	getAlpha	11

5.1.2.6	getEtat	11
5.1.2.7	getEtats	12
5.1.2.8	getTabTransitions	12
5.1.2.9	isDeterministe	12
5.1.2.10	isStandard	12
5.1.2.11	minimise	12
5.1.2.12	produit	13
5.1.2.13	standardise	13
5.1.2.14	supprimeEtat	13
5.1.2.15	supprimeEtat	13
5.1.2.16	supprimerEtatsNonAccessibles	13
5.1.2.17	toDot	14
5.1.3	Member Data Documentation	14
5.1.3.1	etats	14
5.2	choixPointe Class Reference	14
5.3	Ui::choixPointe Class Reference	15
5.4	Ui::CreateAutomate Class Reference	15
5.5	CreateAutomate Class Reference	15
5.6	etat Class Reference	16
5.7	etatLeft Class Reference	17
5.8	Ui::etatLeft Class Reference	18
5.9	Ui::etatRight Class Reference	18
5.10	etatRight Class Reference	18
5.11	Ui::MainWindow Class Reference	19
5.12	MainWindow Class Reference	20
5.13	qt_meta_stringdata_choixPointe_t Struct Reference	21
5.14	qt_meta_stringdata_CreateAutomate_t Struct Reference	21
5.15	qt_meta_stringdata_etatLeft_t Struct Reference	21
5.16	qt_meta_stringdata_etatRight_t Struct Reference	21
5.17	qt_meta_stringdata_MainWindow_t Struct Reference	21
5.18	qt_meta_stringdata_Transition_t Struct Reference	22
5.19	Ui::Transition Class Reference	22
5.20	Transition Class Reference	22
5.21	Ui_choixPointe Class Reference	23
5.22	Ui_CreateAutomate Class Reference	24
5.23	Ui_etatLeft Class Reference	24
5.24	Ui_etatRight Class Reference	25
5.25	Ui_MainWindow Class Reference	26
5.26	Ui_Transition Class Reference	27

6	File Documentation	29
6.1	/home/aaiiighht/Bureau/projetTL/automate-project/automate.h File Reference	29
6.1.1	Detailed Description	29
6.1.2	Function Documentation	29
6.1.2.1	equal	29
6.1.2.2	isFinal	30
Index		31

Chapter 1

Documentation du logiciel Autoroute

1.1 Introduction

En allant dans la section Classes, vous auriez accès à la documentation de l'ensemble des classes. A partir de là, vous pourriez trouver de la doc concernant les attributs et méthodes des classes.

Dans cette page principale, nous verrons comment exécuter le logiciel Autoroute, prendre en main les sources, l'arborescence du projet ainsi que des définitions relatives à la théorie des langages et aux automates.

1.2 Installation du logiciel pour les utilisateurs

Bla Bla Bla

1.3 Pour les développeurs

1.3.1 Etape 1 : Prise en main des sources et execution

Ce logiciel est développé en C++, avec le framework QT5. La manière la plus simple d'accéder aux sources, d'exécuter le programme et de modifier ce logiciel est la suivante :

- installer QT
- créer un dossier dans lequel vous mettrez les 3 dossiers (executable, doc et automate-project)
- dans QT, cliquez sur Open a project puis allez chercher le fichier automate-project/autoroute.pro
- pour lancer le logiciel, cliquez simplement sur la flèche verte

Il vous faudra peut-être configurer dans l'onglet "Projects" l'exécution. Il suffit normalement de préciser le dossier automate/project et d'utiliser les paramètres par défaut.

NB : Vous aurez peut-être un problème de version si vous avez une version supérieure à QT5. Il suffit en général de modifier le nom des bibliothèques. Si cela ne change pas, il vous reste plusieurs solutions :

- aller voir sur le net comment passer le projet de QT5 à la version actuelle de QT
- résoudre les erreurs de compilation (aidez vous du debugger de QT), c'est la solution conseillée.

1.3.2 Etape 2 : Arborescence du projet

- doc/ : vous trouverez ici deux dossiers (html et latex) correspondant à deux formats de la documentation.

Il est possible d'ouvrir ce fichier avec Doxygen et de générer la documentation du programme si vous voulez la modifier. Ce tutoriel est assez bien fait pour prendre en main doxygen : <http://franckh.developpez.com/tutoriels/outils/doxygen/>

- automate-project/ : les sources du programme.

Mieux vaut ne pas y toucher au début, surtout si l'on ne connaît pas QT et modifier le code seulement via QT.

- executable/ : tout les fichiers relatifs aux exécutables

1.4 Définitions

1.4.1 Minimisation d'un automate

1.4.2 Standardisation d'un automate

1.4.3 Produit de deux automates

1.4.4 Détermination de deux automates

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Automate	9
etat	16
QMainWindow	
CreateAutomate	15
MainWindow	20
qt_meta_stringdata_choixPointe_t	21
qt_meta_stringdata_CreateAutomate_t	21
qt_meta_stringdata_etatLeft_t	21
qt_meta_stringdata_etatRight_t	21
qt_meta_stringdata_MainWindow_t	21
qt_meta_stringdata_Transition_t	22
QWidget	
choixPointe	14
etatLeft	17
etatRight	18
Transition	22
Ui_choixPointe	23
Ui::choixPointe	15
Ui_CreateAutomate	24
Ui::CreateAutomate	15
Ui_etatLeft	24
Ui::etatLeft	18
Ui_etatRight	25
Ui::etatRight	18
Ui_MainWindow	26
Ui::MainWindow	19
Ui_Transition	27
Ui::Transition	22

Chapter 3

Class Index

3.1 Class List

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

Automate	9
choixPointe	14
Ui::choixPointe	15
Ui::CreateAutomate	15
CreateAutomate	15
etat	16
etatLeft	17
Ui::etatLeft	18
Ui::etatRight	18
etatRight	18
Ui::MainWindow	19
MainWindow	20
qt_meta_stringdata_choixPointe_t	21
qt_meta_stringdata_CreateAutomate_t	21
qt_meta_stringdata_etatLeft_t	21
qt_meta_stringdata_etatRight_t	21
qt_meta_stringdata_MainWindow_t	21
qt_meta_stringdata_Transition_t	22
Ui::Transition	22
Transition	22
Ui_choixPointe	23
Ui_CreateAutomate	24
Ui_etatLeft	24
Ui_etatRight	25
Ui_MainWindow	26
Ui_Transition	27

Chapter 4

File Index

4.1 File List

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

/home/aaiighht/Bureau/projetTL/automate-project/ automate.h	
Représente un automate, son seul attribut est un vector d'états	29
/home/aaiighht/Bureau/projetTL/automate-project/ choixpointe.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ createautomate.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ etat.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ etatleft.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ etatright.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ mainwindow.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ transition.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ ui_choixpointe.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ ui_createautomate.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ ui_etatleft.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ ui_etatright.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ ui_mainwindow.h	??
/home/aaiighht/Bureau/projetTL/automate-project/ ui_transition.h	??

Chapter 5

Class Documentation

5.1 Automate Class Reference

Public Member Functions

- [Automate](#) ()
Constructeur sans paramètre.
- [Automate](#) (const [Automate](#) &a)
Constructeur.
- [~Automate](#) ()
Destructeur.
- void [ajoutEtat](#) ([etat](#) cible)
Ajout d'un état.
- vector< int > [getTabTransitions](#) ()
Liste les différentes transitions de l'automate.
- bool [isDeterministe](#) ()
Test si un automate est déterministe.
- bool [isStandard](#) ()
Test si un automate est standard.
- [etat](#) * [getEtat](#) (int number)
Retourne un pointeur vers l'état dont le numéro est précisé en paramètre.
- vector< [etat](#) > [getEtats](#) ()
Retourne le vector d'états de l'automate.
- void [ajoutTransition](#) ([etat](#) from, [etat](#) to, int vocab)
Ajoute une transition à un état.
- void [supprimeEtat](#) ([etat](#) cible)
Supprime un état de l'automate.
- int [cible_transition](#) (int etatDepart, int etiq)
Retourne l'état ciblé par une transition.
- void [supprimeEtat](#) ([etat](#) cible, [Automate](#) *a)
Supprime un état de l'automate passé en paramètre.
- void [supprimerEtatsNonAccessibles](#) ([Automate](#) *a)
Supprime les états non accessibles de l'automate passé en paramètre.
- string [toDot](#) ()
Fonction permettant de renvoyer une chaîne à partir de l'automate actuel.
- vector< [Automate](#) > [produit](#) ([Automate](#) A)
Réalise le produit de deux automates.

- `vector< int > getAlpha ()`
Taille de l'alphabet de l'automate.
- `vector< pair< Automate, string > > determine ()`
Réalise la détermination de l'automate.
- `vector< pair< Automate, string > > standardise ()`
Réalise la standardisation de l'automate.
- `vector< pair< Automate, string > > minimise ()`
Réalise la minimisation de l'automate.

Public Attributes

- `vector< etat > etats`

5.1.1 Constructor & Destructor Documentation

5.1.1.1 `Automate::Automate ()`

Constructeur sans paramètre.

Constructeur de la classe [Automate](#), produit un automate vide.

5.1.1.2 `Automate::Automate (const Automate & a)`

Constructeur.

Construit l'automate passé en paramètre

Parameters

<code>a</code>	: automate à construire
----------------	-------------------------

5.1.1.3 `Automate::~~Automate ()`

Destructeur.

Destructeur de la classe [Automate](#)

5.1.2 Member Function Documentation

5.1.2.1 `void Automate::ajoutEtat (etat cible)`

Ajout d'un état.

Methode qui permet d'ajouter un état à l'automate

Parameters

<code>cible</code>	: l'état à ajouter
--------------------	--------------------

5.1.2.2 `void Automate::ajoutTransition (etat from, etat to, int vocab)`

Ajoute une transition à un état.

Ajoute une transition à l'état from en direction de l'état to et portant l'étiquette vocab.

Parameters

<i>from</i>	: l'état de départ de la transition
<i>to</i>	: l'état d'arrivée de la transition
<i>vocab</i>	: le numéro de la transition, son étiquette.

5.1.2.3 `int Automate::cible_transition (int etatDepart, int etiq)`

Retourne l'état ciblé par une transition.

Renvoie le numéro de l'état ciblé par la transition partant de *etatDepart* et portant l'étiquette *etiq*. Cette fonction ne fonctionne que si l'automate est déterministe, elle est utilisée seulement dans la minimisation

Parameters

<i>etatDepart</i>	: le numéro de l'état d'où part la transition
<i>etiq</i>	: l'étiquette de la transition, partant de <i>etatDepart</i>

Returns

le numéro de l'état ciblé par la transition, -1 s'il n'y en a pas

5.1.2.4 `vector< pair< Automate, string > > Automate::determinise ()`

Réalise la détermination de l'automate.

Réalise la détermination de l'automate, et renvoie un vecteur. Dans ce vecteur, chaque élément correspond à une paire : un automate et une chaîne. Chaque élément pair représente en fait une étape dans le processus de détermination. La chaîne est le texte correspondant aux explications et l'automate est l'automate à afficher pendant cette étape.

Returns

le vecteur servant pour la détermination

5.1.2.5 `vector< int > Automate::getAlpha ()`

Taille de l'alphabet de l'automate.

Fonction qui retourne la taille de l'alphabet d'un automate

Returns

un vecteur d'entier, représentant l'ensemble des étiquettes différentes des transitions, c'est-à-dire l'alphabet de l'automate

5.1.2.6 `etat * Automate::getEtat (int number)`

Retourne un pointeur vers l'état dont le numéro est précisé en paramètre.

Récupère le pointeur vers l'état dont le numéro est précisé en paramètre. Récupère cet état dans le vecteur d'états (un attribut de l'automate).

Parameters

<i>number</i>	: le numéro de l'état à retourner
---------------	-----------------------------------

Returns

Un pointeur vers l'état

5.1.2.7 `vector< etat > Automate::getEtats ()`

Retourne le vector d'états de l'automate.

Returns

Le vecteur d'états de l'automate.

5.1.2.8 `vector< int > Automate::getTabTransitions ()`

Liste les différentes transitions de l'automate.

Methode permettant de lister dans un vector d'int, les différentes transitions. [getTabTransitions\(\).size\(\)](#) permet donc de connaitre le nombre de transitions différentes dans l'automate

Returns

un vecteur, chaque entier du vecteur représentant un type de transition

5.1.2.9 `bool Automate::isDeterministe ()`

Test si un automate est déterministe.

Permet de tester si un automate est déterministe (voir définition d'un automate déterministe).

Returns

true si l'automate est déterministe, false sinon

5.1.2.10 `bool Automate::isStandard ()`

Test si un automate est standard.

Permet de tester si un automate est standard (voir définition d'un automate standard).

Returns

true si l'automate est standard, false sinon

5.1.2.11 `vector< pair< Automate, string > > Automate::minimise ()`

Réalise la minimisation de l'automate.

Réalise la minimisation de l'automate, et renvoie un vecteur. Dans ce vecteur, chaque élément correspond à une pair : un automate et une chaîne. Chaque élément pair représente en fait une étape dans le processus de minimisation. La chaîne est le texte correspondant aux explications et l'automate est l'automate à afficher pendant cette étape.

Returns

le vecteur servant pour la minimisation

5.1.2.12 `vector< Automate > Automate::produit (Automate A)`

Réalise le produit de deux automates.

Réalise le produit de deux automates (this et A), et renvoie un vecteur d'automates. Dans ce vecteur, chaque élément correspond à une étape du processus de produit de 2 automates.

Parameters

<i>A</i>	: le produit est réalisé avec cet automate A, passé en paramètre
----------	--

Returns

un vecteur d'automate, chaque automate correspondant à une étape dans le logiciel

5.1.2.13 `vector< pair< Automate, string > > Automate::standardise ()`

Réalise la standardisation de l'automate.

Réalise la standardisation de l'automate, et renvoie un vecteur. Dans ce vecteur, chaque élément correspond à une paire : un automate et une chaîne. Chaque élément pair représente en fait une étape dans le processus de standardisation. La chaîne est le texte correspondant aux explications et l'automate est l'automate à afficher pendant cette étape.

Returns

le vecteur servant pour la standardisation

5.1.2.14 `void Automate::supprimeEtat (etat cible)`

Supprime un état de l'automate.

Supprime l'état, passé en paramètre, de l'automate.

Parameters

<i>cible</i>	: l'état à supprimer
--------------	----------------------

5.1.2.15 `void Automate::supprimeEtat (etat cible, Automate * a)`

Supprime un état de l'automate passé en paramètre.

Supprime l'état cible de l'automate dont le pointeur a est passé en paramètre.

Parameters

<i>cible</i>	: l'état à supprimer
<i>a</i>	: pointeur vers l'automate

5.1.2.16 `void Automate::supprimerEtatsNonAccessibles (Automate * a)`

Supprime les états non accessibles de l'automate passé en paramètre.

Supprime les états non accessibles de l'automate dont le pointeur est passé en paramètre

Parameters

<i>a</i>	: pointeur vers l'automate
----------	----------------------------

5.1.2.17 string Automate::toDot ()

Fonction permettant de renvoyer une chaine à partir de l'automate actuel.

Cette chaine correspond à la représentation de l'automate en graphe dans le langage de description DOT

Returns

La chaine décrivant l'automate, à mettre dans un .dot ensuite

5.1.3 Member Data Documentation**5.1.3.1 vector<etat> Automate::etats**

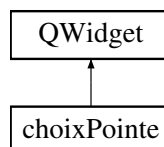
Vecteur des états de l'automate

The documentation for this class was generated from the following files:

- /home/aaiighht/Bureau/projetTL/automate-project/[automate.h](#)
- /home/aaiighht/Bureau/projetTL/automate-project/automate.cpp

5.2 choixPointe Class Reference

Inheritance diagram for choixPointe:

**Public Slots**

- void **sendDad** ()

Signals

- void **add** (int cible, int vocab)

Public Member Functions

- **choixPointe** (QWidget *parent=0)
- void **resetAffichage** ([Automate](#))

Protected Member Functions

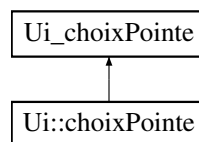
- void **changeEvent** (QEvent *e)

The documentation for this class was generated from the following files:

- /home/aaiighht/Bureau/projetTL/automate-project/choixpointe.h
- /home/aaiighht/Bureau/projetTL/automate-project/choixpointe.cpp
- /home/aaiighht/Bureau/projetTL/automate-project/moc_choixpointe.cpp

5.3 Ui::choixPointe Class Reference

Inheritance diagram for Ui::choixPointe:



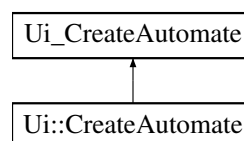
Additional Inherited Members

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

- /home/aaiighht/Bureau/projetTL/automate-project/ui_choixpointe.h

5.4 Ui::CreateAutomate Class Reference

Inheritance diagram for Ui::CreateAutomate:



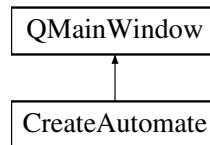
Additional Inherited Members

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

- /home/aaiighht/Bureau/projetTL/automate-project/ui_createautomate.h

5.5 CreateAutomate Class Reference

Inheritance diagram for CreateAutomate:



Public Slots

- void **ajoutEtat** (bool ini=false, bool fina=false)
- void **afficherAutomate** ()
- void **displayRight** (int)
- void **supprimeEtat** (int)
- void **refreshAll** ()
- void **sauvegarder** ()
- void **changeState** (int, bool, bool)

Public Member Functions

- **CreateAutomate** (QWidget *parent=0)
- void **resetAllListChoix** ()
- void **displayAutomate** ()

Public Attributes

- [Automate](#) **a**
- QSvgWidget * **maVue**
- int **actuel**
- vector< [etatLeft](#) * > **left**
- vector< [etatRight](#) * > **right**

Protected Member Functions

- void **changeEvent** (QEvent *e)
- void **adjust** ()

The documentation for this class was generated from the following files:

- /home/aaiighht/Bureau/projetTL/automate-project/createautomate.h
- /home/aaiighht/Bureau/projetTL/automate-project/createautomate.cpp

5.6 etat Class Reference

Public Member Functions

- **etat** (int, bool ini=false, bool fina=false)
- **etat** (const [etat](#) &e)
- int **getNumber** ()
- void **setNumber** (int)
- bool **isFinal** ()
- void **setFinal** (bool)
- bool **isInitial** ()

- void **setInitial** (bool)
- void **ajoutTransition** (etat, int)
- void **supprimeTransition** (etat, int)
- void **renameTransition** (etat, int)
- multimap< int, etat > **getTransitions** ()
- bool **operator==** (etat &) const
- bool **operator!=** (etat &) const
- bool **find_transition** (int etiq, etat e)
- bool **estDansList** (list< etat > liste)
- void **setName** (string)
- string **getName** ()
- string **getNameF** ()
- void **setName** (list< etat > l)

Public Attributes

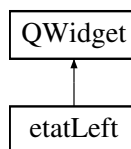
- int **numero**

The documentation for this class was generated from the following files:

- /home/aaiighht/Bureau/projetTL/automate-project/etat.h
- /home/aaiighht/Bureau/projetTL/automate-project/etat.cpp

5.7 etatLeft Class Reference

Inheritance diagram for etatLeft:



Public Slots

- void **sendDad** ()
- void **askForSupress** ()

Signals

- void **selected** (int me)
- void **supress** (int)

Public Member Functions

- **etatLeft** (int, QWidget *parent=0)

Protected Member Functions

- void **changeEvent** (QEvent *e)

Protected Attributes

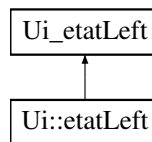
- int **numero**

The documentation for this class was generated from the following files:

- /home/aaiighht/Bureau/projetTL/automate-project/etatleft.h
- /home/aaiighht/Bureau/projetTL/automate-project/etatleft.cpp
- /home/aaiighht/Bureau/projetTL/automate-project/moc_etatleft.cpp

5.8 Ui::etatLeft Class Reference

Inheritance diagram for Ui::etatLeft:



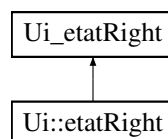
Additional Inherited Members

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

- /home/aaiighht/Bureau/projetTL/automate-project/ui_etatleft.h

5.9 Ui::etatRight Class Reference

Inheritance diagram for Ui::etatRight:



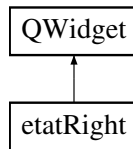
Additional Inherited Members

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

- /home/aaiighht/Bureau/projetTL/automate-project/ui_etatright.h

5.10 etatRight Class Reference

Inheritance diagram for etatRight:



Public Slots

- void **addTransition** (int, int)
- void **eraseTransition** (int, int)
- void **etatChange** ()

Signals

- void **refreshNeeded** (int)
- void **etatChanges** (int, bool, bool)

Public Member Functions

- **etatRight** ([Automate](#) *, int, QWidget *parent=0)
- void **remplirListChoix** ()
- void **addVisualTransition** (int, int)
- void **cleanTrans** ()

Public Attributes

- [choixPointe](#) * **addTrans**
- [Automate](#) * **a**
- int **numero**

Protected Member Functions

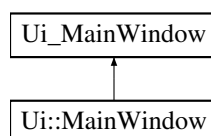
- void **changeEvent** (QEvent *e)

The documentation for this class was generated from the following files:

- /home/aaiighht/Bureau/projetTL/automate-project/etatright.h
- /home/aaiighht/Bureau/projetTL/automate-project/etatright.cpp
- /home/aaiighht/Bureau/projetTL/automate-project/moc_etatright.cpp

5.11 Ui::MainWindow Class Reference

Inheritance diagram for Ui::MainWindow:



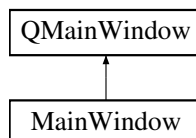
Additional Inherited Members

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

- /home/aaiiighht/Bureau/projetTL/automate-project/ui_mainwindow.h

5.12 MainWindow Class Reference

Inheritance diagram for MainWindow:



Public Slots

- void **openFile** ()
- void **creerAuto** ()
- void **getProduit** ()
- void **getDetermin** ()
- void **getSuivant** ()
- void **getPrecedent** ()
- void **getStandard** ()
- void **getMinimisation** ()
- void **resetUi** ()
- void **test** ()
- void **info** ()

Public Member Functions

- **MainWindow** (QWidget *parent=0)
- void **startLayouting** ()
- void **afficheAutomate** ([Automate](#))
- bool **lireDot** ()
- bool **lireDotB** ()

Public Attributes

- QProcess * **ProcessT**
- QString **program**

Protected Member Functions

- void **changeEvent** (QEvent *e)

The documentation for this class was generated from the following files:

- /home/aaiiighht/Bureau/projetTL/automate-project/mainwindow.h
- /home/aaiiighht/Bureau/projetTL/automate-project/mainwindow.cpp

5.13 qt_meta_stringdata_choixPointe_t Struct Reference

Public Attributes

- QByteArrayData **data** [6]
- char **stringdata** [38]

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

- /home/aaiighht/Bureau/projetTL/automate-project/moc_choixpointe.cpp

5.14 qt_meta_stringdata_CreateAutomate_t Struct Reference

Public Attributes

- QByteArrayData **data** [11]
- char **stringdata** [114]

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

- /home/aaiighht/Bureau/projetTL/automate-project/moc_createautomate.cpp

5.15 qt_meta_stringdata_etatLeft_t Struct Reference

Public Attributes

- QByteArrayData **data** [7]
- char **stringdata** [53]

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

- /home/aaiighht/Bureau/projetTL/automate-project/moc_etatleft.cpp

5.16 qt_meta_stringdata_etatRight_t Struct Reference

Public Attributes

- QByteArrayData **data** [7]
- char **stringdata** [79]

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

- /home/aaiighht/Bureau/projetTL/automate-project/moc_etatright.cpp

5.17 qt_meta_stringdata_MainWindow_t Struct Reference

Public Attributes

- QByteArrayData **data** [13]

- char **stringdata** [125]

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

- /home/aaiighht/Bureau/projetTL/automate-project/moc_mainwindow.cpp

5.18 qt_meta_stringdata_Transition_t Struct Reference

Public Attributes

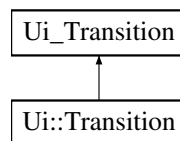
- QByteArrayData **data** [4]
- char **stringdata** [27]

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

- /home/aaiighht/Bureau/projetTL/automate-project/moc_transition.cpp

5.19 Ui::Transition Class Reference

Inheritance diagram for Ui::Transition:



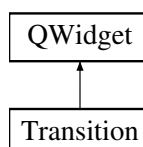
Additional Inherited Members

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

- /home/aaiighht/Bureau/projetTL/automate-project/ui_transition.h

5.20 Transition Class Reference

Inheritance diagram for Transition:



Public Slots

- void **getOff** ()

Signals

- void **eraser** (int, int)

Public Member Functions

- **Transition** (int, int, QWidget *parent=0)

Public Attributes

- int **cible**
- int **vocab**

Protected Member Functions

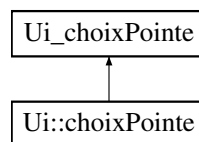
- void **changeEvent** (QEvent *e)

The documentation for this class was generated from the following files:

- /home/aaiighht/Bureau/projetTL/automate-project/transition.h
- /home/aaiighht/Bureau/projetTL/automate-project/moc_transition.cpp
- /home/aaiighht/Bureau/projetTL/automate-project/transition.cpp

5.21 Ui_choixPointe Class Reference

Inheritance diagram for Ui_choixPointe:



Public Member Functions

- void **setupUi** (QWidget *choixPointe)
- void **retranslateUi** (QWidget *choixPointe)

Public Attributes

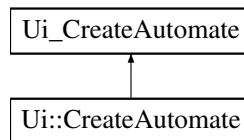
- QVBoxLayout * **verticalLayout_2**
- QFrame * **Widget**
- QHBoxLayout * **horizontalLayout**
- QLabel * **label_3**
- QComboBox * **lesChoix**
- QLabel * **label_4**
- QLineEdit * **lineEdit**
- QPushButton * **pushButton**

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

- /home/aaiighht/Bureau/projetTL/automate-project/ui_choixpointe.h

5.22 Ui_CreateAutomate Class Reference

Inheritance diagram for Ui_CreateAutomate:



Public Member Functions

- void **setupUi** (QMainWindow *[CreateAutomate](#))
- void **retranslateUi** (QMainWindow *[CreateAutomate](#))

Public Attributes

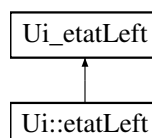
- QAction * **actionFermer**
- QAction * **actionVoir**
- QAction * **actionSauvegarder**
- QWidget * **centralwidget**
- QHBoxLayout * **horizontalLayout**
- QFrame * **frame**
- QVBoxLayout * **etatVert**
- QPushButton * **pushButton**
- QVBoxLayout * **Droite**
- QFrame * **frame1**
- QGridLayout * **etatDroite**
- QScrollArea * **scrollArea**
- QWidget * **vueTomate**
- QMenuBar * **menubar**
- QMenu * **menuFichier**
- QToolBar * **toolBar**

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

- /home/aaiighht/Bureau/projetTL/automate-project/ui_createautomate.h

5.23 Ui_etatLeft Class Reference

Inheritance diagram for Ui_etatLeft:



Public Member Functions

- void **setupUi** (QWidget *[etatLeft](#))
- void **retranslateUi** (QWidget *[etatLeft](#))

Public Attributes

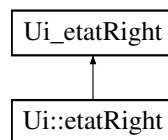
- QGridLayout * **gridLayout**
- QHBoxLayout * **horizontalLayout**
- QSpacerItem * **horizontalSpacer**
- QLabel * **label**
- QSpacerItem * **horizontalSpacer_2**
- QPushButton * **pushButton**
- QSpacerItem * **horizontalSpacer_3**
- QPushButton * **pushButton_2**

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

- /home/aaiiighht/Bureau/projetTL/automate-project/ui_etatleft.h

5.24 Ui_etatRight Class Reference

Inheritance diagram for Ui_etatRight:



Public Member Functions

- void **setupUi** (QWidget *[etatRight](#))
- void **retranslateUi** (QWidget *[etatRight](#))

Public Attributes

- QHBoxLayout * **horizontalLayout**
- QVBoxLayout * **verticalLayout**
- QHBoxLayout * **horizontalLayout_3**
- QSpacerItem * **horizontalSpacer**
- QLabel * **label**
- QSpacerItem * **horizontalSpacer_2**
- QHBoxLayout * **horizontalLayout_2**
- QSpacerItem * **horizontalSpacer_3**
- QCheckBox * **checkBox_2**
- QCheckBox * **checkBox**
- QSpacerItem * **horizontalSpacer_4**
- QHBoxLayout * **Choix**
- QFrame * **frame**
- QVBoxLayout * **ShowChoix**

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

- /home/aaiiighht/Bureau/projetTL/automate-project/ui_etatright.h

5.25 Ui_MainWindow Class Reference

Inheritance diagram for Ui_MainWindow:



Public Member Functions

- void **setupUi** (QMainWindow *[MainWindow](#))
- void **retranslateUi** (QMainWindow *[MainWindow](#))

Public Attributes

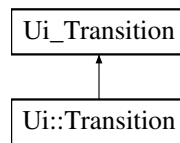
- QAction * **actionFerber**
- QAction * **actionNew**
- QAction * **actionFaireProduit**
- QAction * **actionOuvrir**
- QAction * **actionVoir**
- QAction * **actionTest**
- QAction * **actionClean**
- QAction * **actionInfo**
- QAction * **actionDeterminiser**
- QAction * **actionStandardiser**
- QAction * **actionMinimiser**
- QWidget * **centralWidget**
- QHBoxLayout * **horizontalLayout**
- QVBoxLayout * **verticalLayout**
- QHBoxLayout * **TopLayout**
- QPushButton * **boutonPrec**
- QPushButton * **boutonSuiv**
- QHBoxLayout * **MiddleLayout**
- QScrollArea * **scrollArea_3**
- QWidget * **vue1**
- QScrollArea * **scrollArea_2**
- QWidget * **vue2**
- QHBoxLayout * **LowerLayout**
- QScrollArea * **scrollArea**
- QWidget * **vueTomate**
- QTextEdit * **label**
- QMenuBar * **menuBar**
- QMenu * **menuFichier**
- QToolBar * **toolBar_2**
- QStatusBar * **statusBar**

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

- /home/aaiighht/Bureau/projetTL/automate-project/ui_mainwindow.h

5.26 Ui_Transition Class Reference

Inheritance diagram for Ui_Transition:



Public Member Functions

- void **setupUi** (QWidget * [Transition](#))
- void **retranslateUi** (QWidget * [Transition](#))

Public Attributes

- QHBoxLayout * **horizontalLayout**
- QLabel * **label**
- QSpacerItem * **horizontalSpacer**
- QPushButton * **supress**

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

- /home/aaiighht/Bureau/projetTL/automate-project/ui_transition.h

Chapter 6

File Documentation

6.1 /home/aaiighht/Bureau/projetTL/automate-project/automate.h File Reference

Représente un automate, son seul attribut est un vector d'états.

```
#include <vector>
#include <set>
#include <list>
#include <map>
#include <string>
#include <cstdio>
#include <iostream>
#include <algorithm>
#include "etat.h"
```

Classes

- class [Automate](#)

Functions

- bool [equal](#) (list< [etat](#) > &l1, list< [etat](#) > &l2)
Test l'égalité entre deux listes d'états.
- bool [isFinal](#) (list< [etat](#) > l)
Test si une liste d'états a au moins un état final.

6.1.1 Detailed Description

Représente un automate, son seul attribut est un vector d'états.

6.1.2 Function Documentation

6.1.2.1 bool [equal](#) (list< [etat](#) > & l1, list< [etat](#) > & l2)

Test l'égalité entre deux listes d'états.

Test l'égalité entre deux listes d'états l1 et l2

Returns

true si les listes sont égales, false sinon

6.1.2.2 bool isFinal (list< état > l)

Test si une liste d'états a au moins un état final.

Fonction utilisée seulement pour la détermination

Parameters

	/	: liste d'états testée
--	---	------------------------

Returns

true s'il y a au moins un état final dans la liste d'états, false sinon

Index

- ~Automate
 - Automate, [10](#)
- /home/aaaiighht/Bureau/projetTL/automate-project/automate.h, [29](#)
- ajoutEtat
 - Automate, [10](#)
- ajoutTransition
 - Automate, [10](#)
- Automate, [9](#)
 - ~Automate, [10](#)
 - ajoutEtat, [10](#)
 - ajoutTransition, [10](#)
 - Automate, [10](#)
 - cible_transition, [11](#)
 - determinise, [11](#)
 - etats, [14](#)
 - getAlpha, [11](#)
 - getEtat, [11](#)
 - getEtats, [12](#)
 - getTabTransitions, [12](#)
 - isDeterministe, [12](#)
 - isStandard, [12](#)
 - minimise, [12](#)
 - produit, [12](#)
 - standardise, [13](#)
 - supprimeEtat, [13](#)
 - supprimerEtatsNonAccessibles, [13](#)
 - toDot, [14](#)
- automate.h
 - equal, [29](#)
 - isFinal, [30](#)
- choixPointe, [14](#)
- cible_transition
 - Automate, [11](#)
- CreateAutomate, [15](#)
- determinise
 - Automate, [11](#)
- equal
 - automate.h, [29](#)
- etat, [16](#)
- etatLeft, [17](#)
- etatRight, [18](#)
- etats
 - Automate, [14](#)
- getAlpha
 - Automate, [11](#)
- getEtat
 - Automate, [11](#)
- getEtats
 - Automate, [12](#)
- getTabTransitions
 - Automate, [12](#)
- isDeterministe
 - Automate, [12](#)
- isFinal
 - automate.h, [30](#)
- isStandard
 - Automate, [12](#)
- MainWindow, [20](#)
- minimise
 - Automate, [12](#)
- produit
 - Automate, [12](#)
- qt_meta_stringdata_CreateAutomate_t, [21](#)
- qt_meta_stringdata_MainWindow_t, [21](#)
- qt_meta_stringdata_Transition_t, [22](#)
- qt_meta_stringdata_choixPointe_t, [21](#)
- qt_meta_stringdata_etatLeft_t, [21](#)
- qt_meta_stringdata_etatRight_t, [21](#)
- standardise
 - Automate, [13](#)
- supprimeEtat
 - Automate, [13](#)
- supprimerEtatsNonAccessibles
 - Automate, [13](#)
- toDot
 - Automate, [14](#)
- Transition, [22](#)
- Ui::CreateAutomate, [15](#)
- Ui::MainWindow, [19](#)
- Ui::Transition, [22](#)
- Ui::choixPointe, [15](#)
- Ui::etatLeft, [18](#)
- Ui::etatRight, [18](#)
- Ui_CreateAutomate, [24](#)
- Ui_MainWindow, [26](#)
- Ui_Transition, [27](#)
- Ui_choixPointe, [23](#)
- Ui_etatLeft, [24](#)
- Ui_etatRight, [25](#)