GeradoreReconhecedordeLinguagens

Generated by Doxygen 1.8.6

Wed Jun 17 2015 13:55:40

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 slr.expression.BinaryTree< T > Class Reference

Public Member Functions

- BinaryTree ()
- BinaryTree (BinaryTreeNode< T > root)
- BinaryTreeNode< T > getRoot ()
- void setRoot (BinaryTreeNode< T > root)
- void print ()

3.1.1 Detailed Description

Árvore binária.

Parameters

<T>	

3.1.2 Constructor & Destructor Documentation

3.1.2.1 slr.expression.BinaryTree< T >.BinaryTree ( )

Construtor.

3.1.2.2 slr.expression.BinaryTree< T >.BinaryTree (BinaryTreeNode< T > *root*)

Construtor.

Parameters

root	raiz da árvore.

3.1.3 Member Function Documentation

3.1.3.1 BinaryTreeNode<T> slr.expression.BinaryTree< T >.getRoot ( )

Obter a raiz da árvore.

Returns

nodo raiz da árvore.

3.1.3.2 void slr.expression.BinaryTree< T >.print ( )

Imprimir a árvore.

3.1.3.3 void slr.expression.BinaryTree< T >.setRoot (BinaryTreeNode< T > *root*)

Definir a raiz da árvore.

Parameters

root	nodo raiz da árvore.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/expression/BinaryTree.java

3.2 slr.expression.BinaryTreeNode< T > Class Reference

Public Member Functions

- BinaryTreeNode (T value)
- boolean **equals** (Object o)
- String **toString** ()
- int getNumber ()
- T getValue ()
- void setValue (T value)
- BinaryTreeNode< T > getSeam ()
- void setSeam (BinaryTreeNode< T > seam)
- BinaryTreeNode< T > getLeftNode ()
- void setLeftNode (BinaryTreeNode< T > leftNode)
- BinaryTreeNode< T > getRightNode ()
- void setRightNode (BinaryTreeNode< T > rightNode)
- boolean isVisited ()
- void setVisited (boolean isVisited)
- boolean isLeaf ()
- void print ()
- void sewNode (BinaryTreeNode< T > parent, BinaryTreeNode< T > lastLeftChildParent)
- Set< BinaryTreeNode< T > > getLeaves ()
- Set< BinaryTreeNode< T > > getReachableNodes ()

3.2.1 Detailed Description

Nodo de árvore binária.

Parameters

<T>	

3.2.2 Constructor & Destructor Documentation

3.2.2.1 slr.expression.BinaryTreeNode< T >.BinaryTreeNode (T *value*)

Construtor.

Parameters

value	valor do nodo.

3.2.3 Member Function Documentation

3.2.3.1 Set<BinaryTreeNode<T> > slr.expression.BinaryTreeNode< T >.getLeaves ( )

Obter as folhas a partir desse nodo.

Returns

conjunto de nodos folha.

3.2.3.2 BinaryTreeNode<T> slr.expression.BinaryTreeNode< T >.getLeftNode ( )

Obter o nodo filho da esquerda.

Returns

nodo filho da esquerda.

3.2.3.3 int slr.expression.BinaryTreeNode< T >.getNumber ( )

Obter o número do nodo.

Returns

número.

3.2.3.4 Set<BinaryTreeNode<T> > slr.expression.BinaryTreeNode< T >.getReachableNodes ( )

Obter os nodos alcançáveis a partir desse através das rotinas Descer e Subir.

Returns

conjunto de nodos alcançáveis.

3.2.3.5 BinaryTreeNode<T> slr.expression.BinaryTreeNode< T >.getRightNode ( )

Obter o nodo filho da direita.

Returns

nodo filho da direita.

3.2.3.6 BinaryTreeNode<T> slr.expression.BinaryTreeNode< T >.getSeam ( )

Obter o nodo da costura.

Returns

costura.

3.2.3.7 T slr.expression.BinaryTreeNode< T >.getValue ( )

Obter o valor do nodo.

Returns

valor.

3.2.3.8 boolean slr.expression.BinaryTreeNode< T >.isLeaf ( )

Verificar se o nodo é folha.

Returns

true caso o nodo não possua filhos.

3.2.3.9 boolean slr.expression.BinaryTreeNode< T >.isVisited ( )

Verificar se o nodo foi visitado.

Returns

true caso o nodo já foi visitado.

3.2.3.10 void slr.expression.BinaryTreeNode< T >.print ( )

Imprimir a árvore a partir desse nodo.

3.2.3.11 void slr.expression.BinaryTreeNode< T >.setLeftNode (BinaryTreeNode< T > *leftNode*)

Definir o nodo filho da esquerda.

Parameters

leftNode	nodo filho da esquerda.

3.2.3.12 void slr.expression.BinaryTreeNode< T >.setRightNode (BinaryTreeNode< T > *rightNode*)

Definir o nodo filho da direita.

Parameters

leftNode	nodo filho da direita.

3.2.3.13 void slr.expression.BinaryTreeNode< T >.setSeam (BinaryTreeNode< T > *seam*)

Definir a costura.

Parameters

seam	nodo de costura.

3.2.3.14 void slr.expression.BinaryTreeNode< T >.setValue (T *value*)

Definir o valor do nodo. param value valor do nodo.

3.2.3.15 void slr.expression.BinaryTreeNode< T >.setVisited (boolean *isVisited*)

Definir se o nodo foi visitado.

Parameters

isVisited	true se o nodo foi visitado.

3.2.3.16 void slr.expression.BinaryTreeNode< T >.sewNode (BinaryTreeNode< T > *parent*, BinaryTreeNode< T > *lastLeftChildParent*)

Costurar o nodo.

Parameters

parent	nodo pai.
lastLeftChild-Parent	último nodo pai de um filho da esquerda.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/expression/BinaryTreeNode.java

3.3 slr.automaton.FiniteAutomaton Class Reference

Public Member Functions

- FiniteAutomaton (Set< State > states, State initialState)
- Object **clone** () throws CloneNotSupportedException
- String **toString** ()
- String getName ()
- String[][] toTransitionsTable ()
- void setReferencedAutomaton (String label)
- String getAlphabet ()
- State getInitialState ()
- Set< State > getFinalStates ()
- Set< State > getNotFinalStates ()
- boolean recognize (String entry)
- boolean isEquivalentTo (FiniteAutomaton automaton)

- boolean contains (FiniteAutomaton automaton)
- void complete ()
- void determinize ()
- void minimize ()
- boolean isComplete ()
- boolean isDeterministic ()
- boolean hasMinimalStateSet ()
- boolean isEmpty ()
- boolean isEpsilonFree ()
- FiniteAutomaton complement ()
- FiniteAutomaton union (FiniteAutomaton automaton)
- List< FiniteAutomaton > intersection (FiniteAutomaton automaton)

3.3.1 Detailed Description

Autômato finito.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 slr.automaton.FiniteAutomaton.FiniteAutomaton (Set< State > *states*, State *initialState*)

Construtor.

Parameters

states	conjunto de estados.
initialState	estado inicial do conjunto de estados.

3.3.3 Member Function Documentation

3.3.3.1 FiniteAutomaton slr.automaton.FiniteAutomaton.complement ( )

Calcular o autômato complemento.

Returns

o autômato resultante do complemento.

3.3.3.2 void slr.automaton.FiniteAutomaton.complete ( )

Completar as transições do autômato.

3.3.3.3 boolean slr.automaton.FiniteAutomaton.contains (FiniteAutomaton *automaton*)

Verificar se a linguagem do autômato especificado está contida na linguagem do autômato em questão.

Parameters

automaton	autômato a ser verificado.

Returns

true se T(automaton) está contida na linguagem do autômato.

3.3.3.4 void slr.automaton.FiniteAutomaton.determinize ( )

Determinizar o autômato caso ele não seja determinístico.

3.3.3.5 String slr.automaton.FiniteAutomaton.getAlphabet ( )

Obter o alfabeto de entrada do autômato.

Returns

alfabeto de entrada.

3.3.3.6 Set<State> slr.automaton.FiniteAutomaton.getFinalStates ( )

Obter os estados finais.

Returns

conjunto de estados finais.

3.3.3.7 State slr.automaton.FiniteAutomaton.getInitialState ( )

Obter o estado inicial do autômato.

Returns

estado inicial.

3.3.3.8 String slr.automaton.FiniteAutomaton.getName ( )

Obter o nome do autômato.

Returns

nome do autômato.

3.3.3.9 Set<State> slr.automaton.FiniteAutomaton.getNotFinalStates ( )

Obter os estados não finais.

Returns

conjunto de estados não finais.

3.3.3.10 boolean slr.automaton.FiniteAutomaton.hasMinimalStateSet ( )

Verificar se o autômato contém o menor número de estados.

Returns

true se o autômato possui o menor número de estados possível.

3.3.3.11 List<FiniteAutomaton> slr.automaton.FiniteAutomaton.intersection (FiniteAutomaton *automaton*)

Calcular o autômato da interseção com o autômato especificado.

Parameters

automaton	autômato finito.

Returns

lista de autômatos referentes à interseção.

3.3.3.12 boolean slr.automaton.FiniteAutomaton.isComplete ( )

Verificar se o autômato é completo.

Returns

true se o autômato possui todas as transições definidas.

3.3.3.13 boolean slr.automaton.FiniteAutomaton.isDeterministic ( )

Verificar se o autômato é determinístico.

Returns

true se o autômato é determinístico.

3.3.3.14 boolean slr.automaton.FiniteAutomaton.isEmpty ( )

Verificar se a linguagem do autômato é vazia.

Returns

true se a linguagem do autômato é vazia.

3.3.3.15 boolean slr.automaton.FiniteAutomaton.isEpsilonFree ( )

Verificar se o autômato não possui &-transições.

Returns

true se o autômato não possui &-transições.

3.3.3.16 boolean slr.automaton.FiniteAutomaton.isEquivalentTo (FiniteAutomaton *automaton*)

Verificar se as linguagens dos autômatos são iguais.

Returns

true caso as linguagens sejam iguais.

3.3.3.17 void slr.automaton.FiniteAutomaton.minimize ( )

Minimizar o autômato.

3.3.3.18 boolean slr.automaton.FiniteAutomaton.recognize (String *entry*)

Reconhecer uma entrada qualquer.

Parameters

entry	entrada qualquer.

Returns

true caso a entrada seja uma sentença da linguagem.

3.3.3.19 void slr.automaton.FiniteAutomaton.setReferencedAutomaton (String *label*)

Definir o autômato referenciado por este.

Parameters

label	nome do autômato referenciado.

3.3.3.20 String [ ][ ] slr.automaton.FiniteAutomaton.toTransitionsTable ( )

Obter a tabela de transições do autômato.

Returns

matriz de strings correspondente à tabela de transições.

3.3.3.21 FiniteAutomaton slr.automaton.FiniteAutomaton.union (FiniteAutomaton *automaton*)

Calcular o autômato da união com o autômato especificado.

Parameters

automaton	autômato finito.

Returns

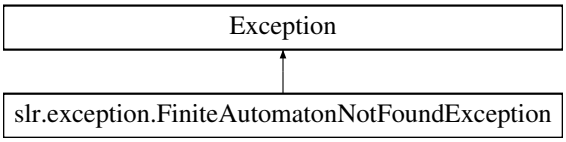
o autômato resultante da união.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/automaton/FiniteAutomaton.java

3.4 slr.exception.FiniteAutomatonNotFoundException Class Reference

Inheritance diagram for slr.exception.FiniteAutomatonNotFoundException:



Public Member Functions

- FiniteAutomatonNotFoundException ()

3.4.1 Detailed Description

Exceção de autômato finito não encontrado.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 slr.exception.FiniteAutomatonNotFoundException.FiniteAutomatonNotFoundException ( )

Construtor.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/exception/FiniteAutomatonNotFoundException.java

3.5 slr.test.FiniteAutomatonTest Class Reference

Public Member Functions

- void setUp () throws Exception
- void **testGetAlphabet** ()
- void **testRecognizeDeterministic** ()
- void **testRecognizeNondeterministic** ()
- void **testRecognizeNondeterministicEpsilon** ()
- void **testRecognizeNondeterministicEpsilon2** ()
- void **testIsEquivalentTo** () throws CloneNotSupportedException
- void **testContains** ()
- void **testContains2** ()
- void **testContains3** ()
- void **testComplete** () throws InvalidTransitionException
- void **testDeterminize** ()
- void **testDeterminize2** ()
- void **testDeterminize3** ()
- void **testDeterminize4** ()
- void **testMinimize** ()
- void **testIsComplete** ()
- void **testIsDeterministic** ()
- void **testIsDeterministic2** ()
- void **testIsDeterministic3** ()
- void **testIsMinimal** ()
- void **testIsEmpty** ()
- void **testUnion** ()
- void **testIntersection** ()
- void **testComplement** ()

3.5.1 Member Function Documentation

3.5.1.1 void slr.test.FiniteAutomatonTest.setUp ( ) throws Exception

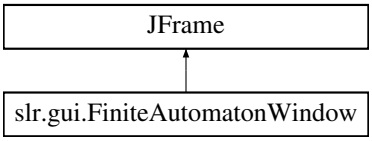
Autômato A = (a,b)$*$ | #a's é divisível por 3 Autômato B = (a,b)$*$ | 'ab' pertence à sentença

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/test/FiniteAutomatonTest.java

3.6 slr.gui.FiniteAutomatonWindow Class Reference

Inheritance diagram for slr.gui.FiniteAutomatonWindow:



Public Member Functions

- **FiniteAutomatonWindow** (String automatonName, String[][] transitionsTable)

3.6.1 Detailed Description

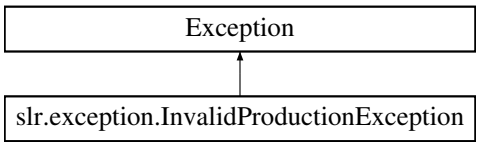
Janela de visualização de autômato.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/gui/FiniteAutomatonWindow.java

3.7 slr.exception.InvalidProductionException Class Reference

Inheritance diagram for slr.exception.InvalidProductionException:



Public Member Functions

- InvalidProductionException ()

3.7.1 Detailed Description

Exceção de produção inválida.

3.7.2 Constructor & Destructor Documentation

3.7.2.1 slr.exception.InvalidProductionException.InvalidProductionException ( )

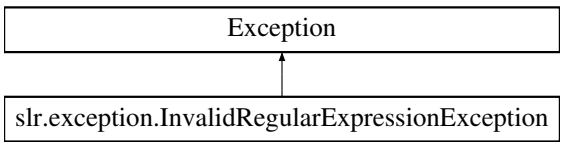
Construtor.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/exception/InvalidProductionException.java

3.8 slr.exception.InvalidRegularExpressionException Class Reference

Inheritance diagram for slr.exception.InvalidRegularExpressionException:



Public Member Functions

- InvalidRegularExpressionException ()

3.8.1 Detailed Description

Exceção de expressão regular inválida.

3.8.2 Constructor & Destructor Documentation

3.8.2.1 slr.exception.InvalidRegularExpressionException.InvalidRegularExpressionException ( )

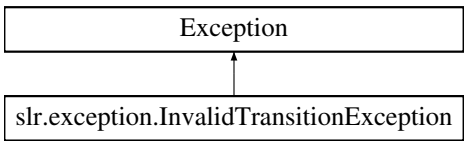
Construtor.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/exception/InvalidRegularExpressionException.java

3.9 slr.exception.InvalidTransitionException Class Reference

Inheritance diagram for slr.exception.InvalidTransitionException:



Public Member Functions

- InvalidTransitionException ()

3.9.1 Detailed Description

Exceção de transição inválida.

3.9.2 Constructor & Destructor Documentation

3.9.2.1 slr.exception.InvalidTransitionException.InvalidTransitionException ( )

Construtor.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/exception/InvalidTransitionException.java

3.10 slr.control.MainController Class Reference

Public Member Functions

- MainController ()
- void execute ()
- void insertRegularDevice (boolean isRegularExpression, String description)
- void removeRegularDevice (String regularDeviceLabel)
- void updateRegularDevice (boolean isRegularExpression, String regularDeviceOldLabel, String regularDeviceDescription)
- String getRegularDeviceLabel (String regularDeviceLabel)
- String[][] getFiniteAutomatonTransitions (String automatonLabel)
- void generateFiniteAutomaton (String regularDeviceLabel)
- void determinizeFiniteAutomaton (String automatonLabel)
- void minimizeFiniteAutomaton (String automatonLabel)
- void complementFiniteAutomaton (String automatonLabel)
- void intersectFiniteAutomata (String automatonLabel1, String automatonLabel2)
- void removeFiniteAutomaton (String automatonLabel)
- boolean areEquivalent (String regularDeviceLabel1, String regularDeviceLabel2)
- void findPatternOccurrences (String regularDeviceLabel, String text)

3.10.1 Detailed Description

Controlador principal.

3.10.2 Constructor & Destructor Documentation

3.10.2.1 slr.control.MainController.MainController ( )

Construtor.

3.10.3 Member Function Documentation

3.10.3.1 boolean slr.control.MainController.areEquivalent (String *regularDeviceLabel1*, String *regularDeviceLabel2*)

Verificar se dois dispositivos regulares são equivalentes.

Parameters

regularDeviceLabel1	nome do dispositivo 1.

regularDevice-Label2	nome do dispositivo 2.

Returns

true caso a linguagem dos dispositivos seja igual.

3.10.3.2 void slr.control.MainController.complementFiniteAutomaton (String *automatonLabel*)

Complementar o autômato finito.

Parameters

automatonLabel	nome do autômato.

3.10.3.3 void slr.control.MainController.determinizeFiniteAutomaton (String *automatonLabel*)

Determinizar o autômato finito.

Parameters

automatonLabel	nome do autômato.

3.10.3.4 void slr.control.MainController.execute ( )

Executar a aplicação.

3.10.3.5 void slr.control.MainController.findPatternOccurrences (String *regularDeviceLabel,* String *text*)

Buscar ocorrência de padrões em texto.

Parameters

regularDevice-Label	nome do dispositivo que define a linguagem.
text	texto a ser analisado.

3.10.3.6 void slr.control.MainController.generateFiniteAutomaton (String *regularDeviceLabel*)

Gerar um autômato finito para um dispositivo regular.

Parameters

regularDevice-Label	nome do dispositivo.

3.10.3.7 String [ ][ ] slr.control.MainController.getFiniteAutomatonTransitions (String *automatonLabel*)

Obter as transições do autômato finito.

Parameters

automatonLabel	nome do autômato.

Returns

tabela de transições do autômato.

3.10.3.8 String slr.control.MainController.getRegularDeviceLabel (String *regularDeviceLabel*)

Obter um dispositivo regular.

Parameters

regularDeviceLabel	nome do dispositivo.

Returns

nome do dispositivo.

3.10.3.9 void slr.control.MainController.insertRegularDevice (boolean *isRegularExpression,* String *description*)

Inserir um dispositivo regular.

Parameters

isRegularExpression	true se é uma expressão regular.
description	descrição textual do dispositivo.

3.10.3.10 void slr.control.MainController.intersectFiniteAutomata (String *automatonLabel1,* String *automatonLabel2*)

Interceptar dois autômatos finitos.

Parameters

automatonLabel1	nome do autômato 1.
automatonLabel2	nome do autômato 2.

3.10.3.11 void slr.control.MainController.minimizeFiniteAutomaton (String *automatonLabel*)

Minimizar o autômato finito.

Parameters

automatonLabel	nome do autômato.

3.10.3.12 void slr.control.MainController.removeFiniteAutomaton (String *automatonLabel*)

Remover autômato finito.

Parameters

automatonLabel	nome do autômato.

3.10.3.13 void slr.control.MainController.removeRegularDevice (String *regularDeviceLabel*)

Remover dispositivo regular.

Parameters

regularDevice-Label	nome do dispositivo.

3.10.3.14 void slr.control.MainController.updateRegularDevice (boolean *isRegularExpression,* String *regularDeviceOldLabel,* String *regularDeviceDescription*)

Atualizar dispositivo regular.

Parameters

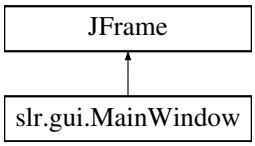
isRegular-Expression	true se é uma expressão regular.
regularDevice-OldLabel	nome antigo do dispositivo.
regularDevice-Description	descrição textual do dispositivo.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/control/MainController.java

3.11 slr.gui.MainWindow Class Reference

Inheritance diagram for slr.gui.MainWindow:



Public Member Functions

- **MainWindow** (final UIController uiController)
- void **insertRegularDevice** (String regularDeviceLabel)
- void **insertFiniteAutomaton** (String automatonLabel)
- void **removeRegularDevice** (String regularDeviceLabel)
- void **removeFiniteAutomaton** (String automatonLabel)
- void **updateRegularDevice** (String regularDeviceOldLabel, String regularDeviceNewLabel)

3.11.1 Detailed Description

Janela principal.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/gui/MainWindow.java

3.12 slr.grammar.ProductionMap Class Reference

Public Member Functions

- ProductionMap ()
- Object **clone** () throws CloneNotSupportedException
- String **toString** ()
- void add (char leftSide, String rightSide)
- void remove (char leftSide, String rightSide) throws InvalidProductionException
- Set< String > get (char leftSide) throws InvalidProductionException
- Set< Character > getNonTerminals ()

3.12.1 Detailed Description

Mapa de produções.

3.12.2 Constructor & Destructor Documentation

3.12.2.1 slr.grammar.ProductionMap.ProductionMap ( )

Construtor.

3.12.3 Member Function Documentation

3.12.3.1 void slr.grammar.ProductionMap.add (char *leftSide,* String *rightSide*)

Adicionar produção.

Parameters

leftSide	símbolo não terminal.
rightSide	símbolo terminal seguido ou não de um símbolo não terminal.

3.12.3.2 Set<String> slr.grammar.ProductionMap.get (char *leftSide*) throws InvalidProductionException

Obter as derivações possíveis a partir de um símbolo.

Parameters

leftSide	símbolo não terminal.

Returns

conjunto de derivações possíveis.

Exceptions

InvalidProductionException	caso não existam produções a partir do símbolo.

3.12.3.3 Set<Character> slr.grammar.ProductionMap.getNonTerminals ( )

Obter o conjunto de símbolos não terminais da gramática.

Returns

conjunto de símbolos não terminais.

3.12.3.4 void slr.grammar.ProductionMap.remove (char *leftSide,* String *rightSide*) throws InvalidProductionException

Remover produção.

Parameters

leftSide	símbolo não terminal.
rightSide	símbolo terminal seguido ou não de um símbolo não terminal.

Exceptions

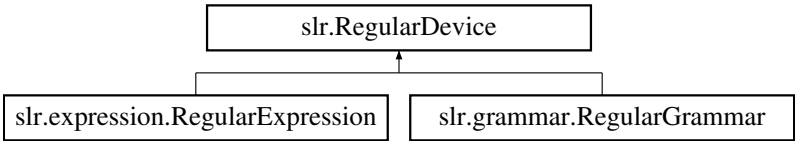
InvalidProductionException	se a produção não existe.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/grammar/ProductionMap.java

3.13 slr.RegularDevice Interface Reference

Inheritance diagram for slr.RegularDevice:



Public Member Functions

- FiniteAutomaton toFiniteAutomaton ()

3.13.1 Detailed Description

Dispositivo regular.

3.13.2 Member Function Documentation

3.13.2.1 FiniteAutomaton slr.RegularDevice.toFiniteAutomaton ( )

Converter o dispositivo em um autômato finito.

Returns

autômato finito equivalente.

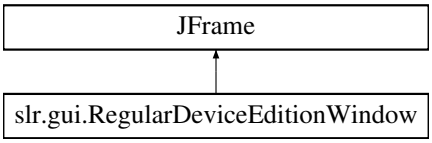
Implemented in slr.expression.RegularExpression, and slr.grammar.RegularGrammar.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/RegularDevice.java

3.14 slr.gui.RegularDeviceEditionWindow Class Reference

Inheritance diagram for slr.gui.RegularDeviceEditionWindow:



Public Member Functions

- **RegularDeviceEditionWindow** (final UIController uiController)
- void **setDeviceType** (boolean isRegularExpression)
- void **setDeviceDescription** (final String text)
- void **setDeviceLabel** (final String label)
- void **setEditionMode** (boolean isEditionMode)

3.14.1 Detailed Description

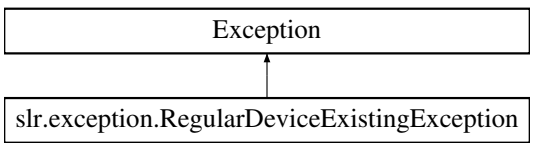
Janela de edição de dispositivos regulares.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/gui/RegularDeviceEditionWindow.java

3.15 slr.exception.RegularDeviceExistingException Class Reference

Inheritance diagram for slr.exception.RegularDeviceExistingException:



Public Member Functions

- RegularDeviceExistingException ()

3.15.1 Detailed Description

Exceção de dispositivo regular existente.

3.15.2 Constructor & Destructor Documentation

3.15.2.1 slr.exception.RegularDeviceExistingException.RegularDeviceExistingException ( )

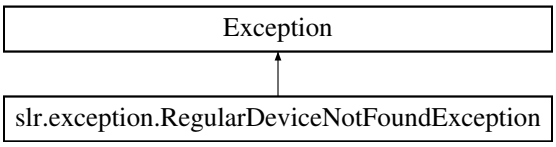
Construtor.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/exception/RegularDeviceExistingException.java

3.16 slr.exception.RegularDeviceNotFoundException Class Reference

Inheritance diagram for slr.exception.RegularDeviceNotFoundException:



Public Member Functions

- RegularDeviceNotFoundException ()

3.16.1 Detailed Description

Exceção de dispositivo regular não encontrado.

3.16.2 Constructor & Destructor Documentation

3.16.2.1 slr.exception.RegularDeviceNotFoundException.RegularDeviceNotFoundException ( )

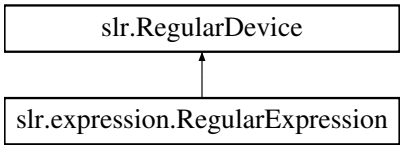
Construtor.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/exception/RegularDeviceNotFoundException.java

3.17 slr.expression.RegularExpression Class Reference

Inheritance diagram for slr.expression.RegularExpression:



Public Member Functions

- RegularExpression (String regularExpression) throws InvalidRegularExpressionException
- String **toString** ()
- void standardize ()
- SyntaxTree getSyntaxTree ()
- String getTerminals ()
- FiniteAutomaton toFiniteAutomaton ()

Static Public Attributes

- static final char **CONCATENATION** = '.'
- static final char **EPSILON** = '&'
- static final char **KLEENE_STAR_CLOSURE** = '*'
- static final char **KLEENE_POSITIVE_CLOSURE** = '+'
- static final char **OR** = '|'
- static final char **OPTIONAL** = '?'
- static final char **PARENTHESIS_OPENING** = '('
- static final char **PARENTHESIS_CLOSING** = ')'
- static final String **ALPHABET** = "abcdefghijklmnopqrstuvwxyz0123456789" + EPSILON

3.17.1 Detailed Description

Expressão regular.

3.17.2 Constructor & Destructor Documentation

3.17.2.1 slr.expression.RegularExpression.RegularExpression (String *regularExpression*) throws InvalidRegularExpressionException

Construtor.

Parameters

regularExpression	expressão regular na forma textual.

Exceptions

InvalidRegularExpressionException	caso a expressão seja inválida.

3.17.3 Member Function Documentation

3.17.3.1 SyntaxTree slr.expression.RegularExpression.getSyntaxTree ( )

Obter a árvore sintática costurada.

Returns

árvore sintática.

3.17.3.2 String slr.expression.RegularExpression.getTerminals ( )

Obter os símbolos terminais da expressão.

Returns

alfabeto.

3.17.3.3 void slr.expression.RegularExpression.standardize ( )

Padronizar a expressão removendo fechamentos positivos e adicionando operadores de concatenação.

3.17.3.4 FiniteAutomaton slr.expression.RegularExpression.toFiniteAutomaton ( )

Converter a expressão regular em um autômato finito.

Returns

autômato finito equivalente.

Implements slr.RegularDevice.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/expression/RegularExpression.java

3.18 slr.expression.RegularExpressionAutomatonBuilder Class Reference

Public Member Functions

- RegularExpressionAutomatonBuilder (RegularExpression regularExpression)
- FiniteAutomaton buildAutomaton ()

3.18.1 Detailed Description

Construtor de autômato finito para expressões regulares.

3.18.2 Constructor & Destructor Documentation

3.18.2.1 slr.expression.RegularExpressionAutomatonBuilder.RegularExpressionAutomatonBuilder (RegularExpression *regularExpression*)

Construtor.

Parameters

regularExpression	expressão regular.

3.18.3 Member Function Documentation

3.18.3.1 FiniteAutomaton slr.expression.RegularExpressionAutomatonBuilder.buildAutomaton ( )

Construir o autômato.

Returns

autômato finito correspondente.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/expression/RegularExpressionAutomatonBuilder.java

3.19 slr.test.RegularExpressionTest Class Reference

Public Member Functions

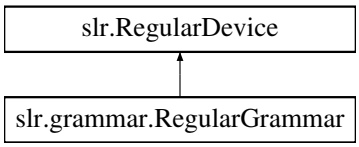
- void **setUp** () throws Exception
- void **testConstructor** () throws InvalidRegularExpressionException
- void **testConstructor2** () throws InvalidRegularExpressionException
- void **testConstructor3** ()
- void **testStandardize** () throws InvalidRegularExpressionException
- void **testGetSyntaxTree** () throws InvalidRegularExpressionException
- void **testToFiniteAutomaton** () throws InvalidRegularExpressionException
- void **testToFiniteAutomaton2** () throws InvalidRegularExpressionException
- void **testToFiniteAutomaton3** () throws InvalidRegularExpressionException
- void **testToFiniteAutomaton4** () throws InvalidRegularExpressionException
- void **testToFiniteAutomaton5** () throws InvalidRegularExpressionException

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/test/RegularExpressionTest.java

3.20 slr.grammar.RegularGrammar Class Reference

Inheritance diagram for slr.grammar.RegularGrammar:



Public Member Functions

- RegularGrammar (String productions) throws InvalidProductionException
- RegularGrammar (ProductionMap productions, char initialSymbol) throws InvalidProductionException
- Object **clone** () throws CloneNotSupportedException
- String **toString** ()
- char getInitialSymbol ()
- FiniteAutomaton toFiniteAutomaton ()

Static Public Attributes

- static final String **DERIVATION** = "->"
- static final String **TERMINALS** = "abcdefghijklmnopqrstuvwxyz0123456789" + RegularExpression.EPSILON
- static final String **NONTERMINALS** = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

3.20.1 Detailed Description

Gramática regular.

3.20.2 Constructor & Destructor Documentation

3.20.2.1 slr.grammar.RegularGrammar.RegularGrammar (String *productions*) throws InvalidProductionException

Construtor.

Parameters

productions	produções na forma textual.

Exceptions

InvalidProductionException	caso as produções sejam inválidas.

3.20.2.2 slr.grammar.RegularGrammar.RegularGrammar (ProductionMap *productions*, char *initialSymbol*) throws InvalidProductionException

Construtor.

Parameters

productions	mapa de produções.
initialSymbol	símbolo inicial da gramática.

Exceptions

InvalidProductionException	caso as produções sejam inválidas.

3.20.3 Member Function Documentation

3.20.3.1 char slr.grammar.RegularGrammar.getInitialSymbol ( )

Obter o símbolo inicial.

Returns

símbolo inicial da gramática.

3.20.3.2 FiniteAutomaton slr.grammar.RegularGrammar.toFiniteAutomaton ( )

Converter a gramática em um autômato finito.

Returns

autômato finito equivalente.

Implements slr.RegularDevice.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/grammar/RegularGrammar.java

3.21 slr.test.RegularGrammarTest Class Reference

Public Member Functions

- void **setUp** () throws InvalidProductionException
- void **testConstructor** () throws InvalidProductionException
- void **testConstructor2** () throws InvalidProductionException
- void **testConstructor3** () throws InvalidProductionException
- void **testConstructor4** () throws InvalidProductionException
- void **testConstructor5** () throws InvalidProductionException
- void **testConstructor6** () throws InvalidProductionException
- void **testToString** ()
- void **testToFiniteAutomaton** () throws InvalidProductionException
- void **testToFiniteAutomaton2** () throws InvalidProductionException

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/test/RegularGrammarTest.java

3.22 slr.control.RegularLanguageController Class Reference

Public Member Functions

- RegularLanguageController ()
- String insertRegularGrammar (String productions) throws RegularDeviceExistingException, InvalidProductionException
- String insertRegularExpression (String regularExpression) throws RegularDeviceExistingException, InvalidRegularExpressionException
- String generateFiniteAutomaton (String regularDeviceLabel) throws RegularDeviceNotFoundException
- void removeRegularDevice (String regularDeviceLabel)
- void removeFiniteAutomaton (String automatonLabel)
- FiniteAutomaton getFiniteAutomaton (String automatonLabel) throws FiniteAutomatonNotFoundException
- String[][] getFiniteAutomatonTransitionTable (String automatonLabel) throws FiniteAutomatonNotFoundException
- String getRegularDeviceTextForm (String regularDeviceLabel) throws RegularDeviceNotFoundException
- Set< String > findPatternOccurrences (FiniteAutomaton automaton, String text)
- boolean finiteAutomataAreEquivalent (FiniteAutomaton fa1, FiniteAutomaton fa2)
- List< String > determinizeFiniteAutomaton (FiniteAutomaton automaton) throws Exception
- List< String > minimizeFiniteAutomaton (FiniteAutomaton automaton) throws Exception
- List< String > complementFiniteAutomaton (FiniteAutomaton automaton)
- List< String > intersectFiniteAutomaton (FiniteAutomaton automaton1, FiniteAutomaton automaton2)

3.22.1 Detailed Description

Controlador de linguagens regulares.

3.22.2 Constructor & Destructor Documentation

3.22.2.1 slr.control.RegularLanguageController.RegularLanguageController ( )

Construtor.

3.22.3 Member Function Documentation

3.22.3.1 List<String> slr.control.RegularLanguageController.complementFiniteAutomaton (FiniteAutomaton *automaton*)

Complementar o autômato finito.

Parameters

automaton	autômato.

Returns

lista de autômatos correspondentes ao complemento.

3.22.3.2 List<String> slr.control.RegularLanguageController.determinizeFiniteAutomaton (FiniteAutomaton *automaton*) throws Exception

Determinizar o autômato finito.

Parameters

automaton	autômato.

Returns

lista de autômatos correspondentes à determinização.

3.22.3.3 Set<String> slr.control.RegularLanguageController.findPatternOccurrences (FiniteAutomaton *automaton,* String *text*)

Buscar ocorrência de padrões em texto.

Parameters

automaton	autômato.
text	texto a ser analisado.

Returns

conjunto de padrões encontrados.

3.22.3.4 boolean slr.control.RegularLanguageController.finiteAutomataAreEquivalent (FiniteAutomaton *fa1,* FiniteAutomaton *fa2*)

Verificar se a linguagem de dois autômatos é igual.

Parameters

fa1	autômato 1.
fa2	autômato 2.

Returns

true caso a linguagem seja igual.

3.22.3.5 String slr.control.RegularLanguageController.generateFiniteAutomaton (String *regularDeviceLabel*) throws RegularDeviceNotFoundException

Gerar um autômato finito para um dispositivo regular.

Parameters

regularDeviceLabel	nome do dispositivo.

Exceptions

RegularDeviceNotFoundException	caso o dispositivo regular não seja encontrado.

3.22.3.6 FiniteAutomaton slr.control.RegularLanguageController.getFiniteAutomaton (String *automatonLabel*) throws FiniteAutomatonNotFoundException

Obter o autômato.

Parameters

automatonLabel	nome do autômato.

Returns

autômato finito.

Exceptions

FiniteAutomatonNotFoundException	caso o autômato não seja encontrado.

3.22.3.7 String [ ][ ] slr.control.RegularLanguageController.getFiniteAutomatonTransitionTable (String *automatonLabel*) throws FiniteAutomatonNotFoundException

Obter a tabela de transições do autômato.

Parameters

automatonLabel	nome do autômato.

Returns

tabela de transições.

Exceptions

FiniteAutomatonNotFoundException	caso o autômato não seja encontrado.

3.22.3.8 String slr.control.RegularLanguageController.getRegularDeviceTextForm (String *regularDeviceLabel*) throws RegularDeviceNotFoundException

Obter a forma textual de um dispositivo regular.

Parameters

regularDeviceLabel	dispositivo regular.

Returns

forma textual do dispositivo.

Exceptions

RegularDeviceNotFound-Exception	caso o dispositivo não seja encontrado.

3.22.3.9 String slr.control.RegularLanguageController.insertRegularExpression (String *regularExpression*) throws RegularDeviceExistingException, InvalidRegularExpressionException

Inserir uma expressão regular.

Parameters

regular-Expression	descrição textual da expressão.

Exceptions

RegularDeviceExisting-Exception	caso a expressão já exista.
InvalidRegularExpression-Exception	caso a expressão seja inválida.

3.22.3.10 String slr.control.RegularLanguageController.insertRegularGrammar (String *productions*) throws RegularDeviceExistingException, InvalidProductionException

Inserir uma gramática regular.

Parameters

productions	produções da gramática.

Exceptions

RegularDeviceExisting-Exception	caso a gramática já exista.
InvalidProductionException	caso o conjunto de produções seja inválido.

3.22.3.11 List<String> slr.control.RegularLanguageController.intersectFiniteAutomaton (FiniteAutomaton *automaton1*, FiniteAutomaton *automaton2*)

Interceptar dois autômatos finitos.

Parameters

automaton1	autômato 1.
automaton2	autômato 2.

Returns

lista de autômatos correspondentes à interseção.

3.22.3.12 List<String> slr.control.RegularLanguageController.minimizeFiniteAutomaton (FiniteAutomaton *automaton*) throws Exception

Minimizar o autômato finito.

Parameters

automaton	autômato.

Returns

lista de autômatos correspondentes à minimização.

3.22.3.13 void slr.control.RegularLanguageController.removeFiniteAutomaton (String *automatonLabel*)

Remover um autômato finito.

Parameters

automatonLabel	nome do autômato.

3.22.3.14 void slr.control.RegularLanguageController.removeRegularDevice (String *regularDeviceLabel*)

Remover um dispositivo regular.

Parameters

regularDevice-Label	nome do dispositivo regular.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/control/RegularLanguageController.java

3.23 slr.SLRApp Class Reference

Static Public Member Functions

- static void main (String[] args) throws InvalidProductionException

3.23.1 Detailed Description

Sistema de linguagens regulares.

3.23.2 Member Function Documentation

3.23.2.1 static void slr.SLRApp.main (String[] *args*) throws InvalidProductionException [static]

Método principal.

Parameters

args	argumentos.

Exceptions

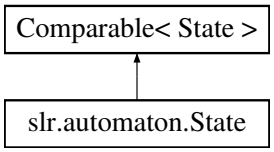
InvalidProductionException	

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/SLRApp.java

3.24 slr.automaton.State Class Reference

Inheritance diagram for slr.automaton.State:



Public Member Functions

- State (String name, boolean isFinal, TransitionMap transitions)
- Object **clone** () throws CloneNotSupportedException
- boolean **equals** (Object obj)
- int **compareTo** (State o)
- void setIsFinal (boolean isFinal)
- void setName (String name)
- String getName ()
- boolean isFinal ()
- Set< State > transit (char symbol) throws InvalidTransitionException
- Set< State > getReachableStates ()
- Set< State > getEpsilonClosure ()
- TransitionMap getTransitionMap ()

3.24.1 Detailed Description

Estado.

3.24.2 Constructor & Destructor Documentation

3.24.2.1 slr.automaton.State.State (String *name,* boolean *isFinal,* TransitionMap *transitions*)

Construtor.

Parameters

name	nome do estado.
isFinal	true caso o estado seja final.
transitions	transições do estado.

3.24.3 Member Function Documentation

3.24.3.1 Set<State> slr.automaton.State.getEpsilonClosure ( )

Obter o Epsilon fecho do estado.

Returns

conjunto de estados alcançáveis por Epsilon.

3.24.3.2 String slr.automaton.State.getName ( )

Obter o nome do estado.

Returns

nome do estado.

3.24.3.3 Set<State> slr.automaton.State.getReachableStates ( )

Obter os estados alcançáveis a partir deste.

Returns

conjunto de estados alcançáveis.

3.24.3.4 TransitionMap slr.automaton.State.getTransitionMap ( )

Obter o mapa de transições do estado.

Returns

mapa de transições.

3.24.3.5 boolean slr.automaton.State.isFinal ( )

Verificar se o estado é final.

Returns

true caso o estado seja final.

3.24.3.6 void slr.automaton.State.setIsFinal (boolean *isFinal*)

Definir se o estado é final.

Parameters

isFinal	true se o estado é final.

3.24.3.7 void slr.automaton.State.setName (String *name*)

Definir o nome do estado.

Parameters

name	nome do estado.

3.24.3.8 Set<State> slr.automaton.State.transit (char *symbol*) throws InvalidTransitionException

Transitar do estado para o(s) próximo(s) pelo símbolo.

Parameters

symbol	símbolo de entrada da transição.

Returns

conjunto de estados de destino da transição.

Exceptions

InvalidTransitionException	se não há uma transição pelo símbolo de entrada especificado.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/automaton/State.java

3.25 slr.test.StateTest Class Reference

Public Member Functions

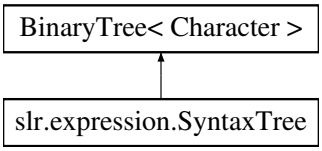
- void **setUp** () throws Exception
- void **testGetName** ()
- void **testIsFinal** ()
- void **testTransitSingle** ()
- void **testTransitMultiple** ()
- void **testTransitInvalid** ()
- void **testGetReachableStates** ()
- void **testEqualsObject** ()

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/test/StateTest.java

3.26 slr.expression.SyntaxTree Class Reference

Inheritance diagram for slr.expression.SyntaxTree:



Public Member Functions

- SyntaxTree (RegularExpression regex)
- Set< BinaryTreeNode< Character > > getLeaves ()

3.26.1 Detailed Description

Árvore sintática.

3.26.2 Constructor & Destructor Documentation

3.26.2.1 slr.expression.SyntaxTree.SyntaxTree (RegularExpression *regex*)

Construtor.

Parameters

regex	expressão regular.

3.26.3 Member Function Documentation

3.26.3.1 Set<BinaryTreeNode<Character> > slr.expression.SyntaxTree.getLeaves ( )

Obter os nodos folhas da árvore.

Returns

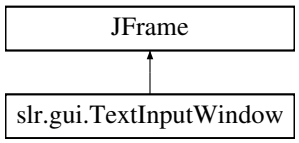
conjunto de folhas.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/expression/SyntaxTree.java

3.27 slr.gui.TextInputWindow Class Reference

Inheritance diagram for slr.gui.TextInputWindow:



Public Member Functions

- **TextInputWindow** (UIController uiController)
- void **setRegularExpression** (String regex)

3.27.1 Detailed Description

Janela de inserção de texto para busca de padrões.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/gui/TextInputWindow.java

3.28 slr.automaton.TransitionMap Class Reference

Public Member Functions

- TransitionMap ()
- Object **clone** () throws CloneNotSupportedException
- void add (final char symbol, final State targetState)
- void remove (final char symbol, final State targetState) throws InvalidTransitionException
- void replaceTargets (final State target, final State newTarget)
- Set< State > get (final char symbol) throws InvalidTransitionException
- Set< State > getTargetStates ()
- Map< Character, Set< State > > getMap ()

3.28.1 Detailed Description

Mapa de transições.

3.28.2 Constructor & Destructor Documentation

3.28.2.1 slr.automaton.TransitionMap.TransitionMap ( )

Construtor.

3.28.3 Member Function Documentation

3.28.3.1 void slr.automaton.TransitionMap.add (final char *symbol*, final State *targetState*)

Adicionar transição.

Parameters

symbol	símbolo de transição.
targetState	estado de destino da transição.

3.28.3.2 Set<State> slr.automaton.TransitionMap.get (final char *symbol*) throws InvalidTransitionException

Obter as transições possíveis por um símbolo.

Parameters

symbol	símbolo de transição.

Returns

conjunto de estados de destino.

Exceptions

InvalidTransitionException	caso não exista transição pelo símbolo.

3.28.3.3 Map<Character, Set<State> > slr.automaton.TransitionMap.getMap ( )

Obter as transições.

Returns

mapa de transições.

3.28.3.4 Set<State> slr.automaton.TransitionMap.getTargetStates ( )

Obter todos os estados para os quais existe uma transição.

Returns

conjunto de estados.

3.28.3.5 void slr.automaton.TransitionMap.remove (final char *symbol,* final State *targetState*) throws InvalidTransitionException

Remover transição.

Parameters

symbol	símbolo de transição.
targetState	estado de destino da transição.

Exceptions

InvalidTransitionException	se a transição não existe.

3.28.3.6 void slr.automaton.TransitionMap.replaceTargets (final State *target,* final State *newTarget*)

Substituir os estados de destino pelo estado especificado.

Parameters

target	estado a ser substituído.
newTarget	novo estado.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/automaton/TransitionMap.java

3.29 slr.control.UIController Class Reference

Public Member Functions

- UIController (MainController mainController)
- void showMainWindow ()
- void showRegularDeviceEditionWindow ()
- void showRegularDeviceEditionWindow (boolean isRegularExpression, String regularDeviceLabel)
- void showFiniteAutomatonWindow (String automatonLabel)
- void showTextInputWindow (String regularExpressionLabel)
- void disposeRegularDeviceEditionWindow ()
- void showErrorMessage (String message)
- void showInformationMessage (String message)
- void insertRegularDeviceToList (String regularDeviceLabel)
- void removeRegularDeviceFromList (String regularDeviceLabel)
- void updateRegularDeviceFromList (String regularDeviceOldLabel, String regularDeviceNewLabel)

- void insertRegularDevice (boolean isRegularExpression, String description)
- void removeRegularDevice (String regularDeviceLabel)
- void updateRegularDevice (boolean isRegularExpression, String regularDeviceOldLabel, String regularDeviceDescription)
- void insertFiniteAutomatonToList (String automatonLabel)
- void removeFiniteAutomatonFromList (String automatonLabel)
- void removeFiniteAutomaton (String automatonLabel)
- void generateFiniteAutomaton (String regularDeviceLabel)
- String[][] getFiniteAutomatonTransitions (String automatonLabel)
- void determinizeFiniteAutomaton (String automatonLabel)
- void minimizeFiniteAutomaton (String automatonLabel)
- void complementFiniteAutomaton (String automatonLabel)
- void intersectFiniteAutomata (String automatonLabel1, String automatonLabel2)
- boolean areEquivalent (String regularDeviceLabel1, String regularDeviceLabel2)
- void findPatternOccurrences (String regularDeviceLabel, String text)

3.29.1 Detailed Description

Controlador da interface gráfica.

3.29.2 Constructor & Destructor Documentation

3.29.2.1 slr.control.UIController.UIController (MainController *mainController*)

Construtor

Parameters

mainController	controlador principal.

3.29.3 Member Function Documentation

3.29.3.1 boolean slr.control.UIController.areEquivalent (String *regularDeviceLabel1*, String *regularDeviceLabel2*)

Verificar se dois dispositivos são equivalentes

Parameters

regularDeviceLabel1	nome do dispositivo regular 1.
regularDeviceLabel2	nome do dispositivo regular 2.

Returns

true caso a linguagem seja igual.

3.29.3.2 void slr.control.UIController.complementFiniteAutomaton (String *automatonLabel*)

Complementar um autômato.

Parameters

automatonLabel	nome do autômato.

3.29.3.3 void slr.control.UIController.determinizeFiniteAutomaton (String *automatonLabel*)

Determinizar um autômato.

Parameters

automatonLabel	nome do autômato.

3.29.3.4 void slr.control.UIController.disposeRegularDeviceEditionWindow ( )

Fechar janela de inserção/edição de dispositivos regulares.

3.29.3.5 void slr.control.UIController.findPatternOccurrences (String *regularDeviceLabel*, String *text*)

Buscar ocorrências de padrão em texto.

Parameters

regularDeviceLabel	nome do dispositivo.
text	texto a ser analisado.

3.29.3.6 void slr.control.UIController.generateFiniteAutomaton (String *regularDeviceLabel*)

Gerar autômato finito.

Parameters

regularDeviceLabel	nome do dispositivo regular.

3.29.3.7 String [ ][ ] slr.control.UIController.getFiniteAutomatonTransitions (String *automatonLabel*)

Obter transições do autômato finito.

Parameters

automatonLabel	nome do autômato.

Returns

tabela de transições do autômato.

3.29.3.8 void slr.control.UIController.insertFiniteAutomatonToList (String *automatonLabel*)

Inserir autômato finito na lista de autômatos.

Parameters

automatonLabel	nome do autômato.

3.29.3.9 void slr.control.UIController.insertRegularDevice (boolean *isRegularExpression,* String *description*)

Inserir dispositivo regular.

Parameters

isRegular-Expression	true se é expressão regular.
description	descrição textual do dispositivo.

3.29.3.10 void slr.control.UIController.insertRegularDeviceToList (String *regularDeviceLabel*)

Inserir dispositivo regular na lista de dispositivos.

Parameters

regularDevice-Label	nome do dispositivo.

3.29.3.11 void slr.control.UIController.intersectFiniteAutomata (String *automatonLabel1,* String *automatonLabel2*)

Interceptar dois autômatos.

Parameters

automaton-Label1	nome do autômato 1.
automaton-Label2	nome do autômato 2.

3.29.3.12 void slr.control.UIController.minimizeFiniteAutomaton (String *automatonLabel*)

Minimizar um autômato.

Parameters

automatonLabel	nome do autômato.

3.29.3.13 void slr.control.UIController.removeFiniteAutomaton (String *automatonLabel*)

Remover autômato finito.

Parameters

automatonLabel	nome do autômato.

3.29.3.14 void slr.control.UIController.removeFiniteAutomatonFromList (String *automatonLabel*)

Remover autômato finito da lista de autômatos.

Parameters

automatonLabel	nome do autômato.

3.29.3.15 void slr.control.UIController.removeRegularDevice (String *regularDeviceLabel*)

Remover dispositivo regular.

Parameters

regularDevice-Label	nome do dispositivo.

3.29.3.16 void slr.control.UIController.removeRegularDeviceFromList (String *regularDeviceLabel*)

Remover dispositivo regular da lista de dispositivos.

Parameters

regularDevice-Label	nome do dispositivo.

3.29.3.17 void slr.control.UIController.showErrorMessage (String *message*)

Exibir mensagem de erro.

Parameters

message	mensagem.

3.29.3.18 void slr.control.UIController.showFiniteAutomatonWindow (String *automatonLabel*)

Exibir a janela de visualização de autômato finito.

Parameters

automatonLabel	nome do autômato.

3.29.3.19 void slr.control.UIController.showInformationMessage (String *message*)

Exibir mensagem informativa.

Parameters

message	mensagem.

3.29.3.20 void slr.control.UIController.showMainWindow ( )

Exibir a janela principal.

3.29.3.21 void slr.control.UIController.showRegularDeviceEditionWindow ( )

Exibir a janela de inserção/edição de dispositivos regulares.

3.29.3.22 void slr.control.UIController.showRegularDeviceEditionWindow (boolean *isRegularExpression,* String *regularDeviceLabel*)

Exibir a janela de inserção/edição de dispositivos regulares.

Parameters

isRegularExpression	true se é expressão regular.
regularDeviceLabel	nome do dispositivo regular.

3.29.3.23 void slr.control.UIController.showTextInputWindow (String *regularExpressionLabel*)

Exibir a janela de busca de padrões em texto.

Parameters

regularExpressionLabel	nome da expressão regular.

3.29.3.24 void slr.control.UIController.updateRegularDevice (boolean *isRegularExpression,* String *regularDeviceOldLabel,* String *regularDeviceDescription*)

Atualizar dispositivo regular.

Parameters

isRegularExpression	true se é expressão regular.
regularDeviceOldLabel	nome antigo do dispositivo.
regularDeviceNewLabel	novo nome do dispositivo.

3.29.3.25 void slr.control.UIController.updateRegularDeviceFromList (String *regularDeviceOldLabel,* String *regularDeviceNewLabel*)

Atualizar dispositivo regular na lista de dispositivos.

Parameters

regularDeviceOldLabel	nome antigo do dispositivo.
regularDeviceNewLabel	novo nome do dispositivo.

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

- /home/lucas/git/LF-SLR/LF-SLR/src/slr/control/UIController.java