SimpleGraph

Generated by Doxygen 1.8.13

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

File Index

3.1 File List

Here is a list of all files with brief descriptions:

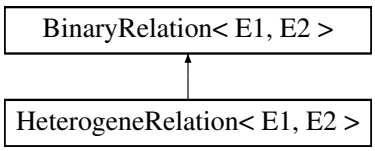
Chapter 4

Class Documentation

4.1 BinaryRelation< E1, E2 > Class Template Reference

`#include <binary_relation.h>`

Inheritance diagram for BinaryRelation< E1, E2 >:



Public Member Functions

- BinaryRelation ()
- BinaryRelation (const BinaryRelation< E1, E2 > &binrel)
- BinaryRelation (const E1 &inputset, const E2 &targetset)
- ~BinaryRelation ()
- bool add (E1 first, E2 second)
- bool remove (E1 first, E2 second)
- bool remove_pair (E1 first, E2 second)
- bool remove_from_pretarget_set (E1 first)
- bool remove_from_target_set (E2 second)
- template<typename T >
  bool check_and_erase (std::set< T > templ_set, T key)
- BinaryRelation< E2, E1 > transposition ()
- template<typename E3 >
  BinaryRelation< E1, E3 > product (BinaryRelation< E2, E3 > relation)
- std::set< E1 > get_input_set ()
- std::set< E2 > get_target_set ()
- std::set< Pair< E1, E2 > > get_pair_set ()
- void set_input_set (const std::set< E1 > &input)
- void set_target_set (const std::set< E2 > &target)
- void set_pair_set (const std::set< Pair< E1, E2 > > &pairs)
- bool operator== (const BinaryRelation< E1, E2 > &comp_rel)

4.1.1 Constructor & Destructor Documentation

4.1.1.1 BinaryRelation() [1/3]

```
template<typename E1, typename E2>
BinaryRelation< E1, E2 >::BinaryRelation ( )  [inline]
```

4.1.1.2 BinaryRelation() [2/3]

```
template<typename E1, typename E2>
BinaryRelation< E1, E2 >::BinaryRelation (
            const BinaryRelation< E1, E2 > & binrel )  [inline]
```

4.1.1.3 BinaryRelation() [3/3]

```
template<typename E1, typename E2>
BinaryRelation< E1, E2 >::BinaryRelation (
            const E1 & inputset,
            const E2 & targetset )  [inline]
```

4.1.1.4 ~BinaryRelation()

```
template<typename E1, typename E2>
BinaryRelation< E1, E2 >::~BinaryRelation ( )  [inline]
```

4.1.2 Member Function Documentation

4.1.2.1 add()

```
template<typename E1, typename E2>
bool BinaryRelation< E1, E2 >::add (
            E1 first,
            E2 second )  [inline]
```

4.1.2.2 check_and_erase()

```
template<typename E1, typename E2>
template<typename T >
bool BinaryRelation< E1, E2 >::check_and_erase (
            std::set< T > templ_set,
            T key )  [inline]
```

4.1.2.3 get_input_set()

```
template<typename E1, typename E2>
std::set<E1> BinaryRelation< E1, E2 >::get_input_set ( )  [inline]
```

4.1.2.4 get_pair_set()

```
template<typename E1, typename E2>
std::set<Pair<E1, E2> > BinaryRelation< E1, E2 >::get_pair_set ( )  [inline]
```

4.1.2.5 get_target_set()

```
template<typename E1, typename E2>
std::set<E2> BinaryRelation< E1, E2 >::get_target_set ( )  [inline]
```

4.1.2.6 operator==()

```
template<typename E1, typename E2>
bool BinaryRelation< E1, E2 >::operator== (
            const BinaryRelation< E1, E2 > & comp_rel )  [inline]
```

4.1.2.7 product()

```
template<typename E1, typename E2>
template<typename E3 >
BinaryRelation<E1, E3> BinaryRelation< E1, E2 >::product (
            BinaryRelation< E2, E3 > relation )  [inline]
```

4.1.2.8 remove()

```
template<typename E1, typename E2>
bool BinaryRelation< E1, E2 >::remove (
            E1 first,
            E2 second )  [inline]
```

4.1.2.9 remove_from_pretarget_set()

```
template<typename E1, typename E2>
bool BinaryRelation< E1, E2 >::remove_from_pretarget_set (
            E1 first )  [inline]
```

4.1.2.10 remove_from_target_set()

```
template<typename E1, typename E2>
bool BinaryRelation< E1, E2 >::remove_from_target_set (
            E2 second )  [inline]
```

4.1.2.11 remove_pair()

```
template<typename E1, typename E2>
bool BinaryRelation< E1, E2 >::remove_pair (
            E1 first,
            E2 second )  [inline]
```

4.1.2.12 set_input_set()

```
template<typename E1, typename E2>
void BinaryRelation< E1, E2 >::set_input_set (
            const std::set< E1 > & input )  [inline]
```

4.1.2.13 set_pair_set()

```
template<typename E1, typename E2>
void BinaryRelation< E1, E2 >::set_pair_set (
            const std::set< Pair< E1, E2 > > & pairs )  [inline]
```

4.1.2.14 set_target_set()

```
template<typename E1, typename E2>
void BinaryRelation< E1, E2 >::set_target_set (
             const std::set< E2 > & target )  [inline]
```

4.1.2.15 transposition()

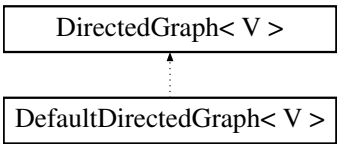
```
template<typename E1, typename E2>
BinaryRelation<E2, E1> BinaryRelation< E1, E2 >::transposition ( )  [inline]
```

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

- src/relation/include/binary_relation.h

4.2 DefaultDirectedGraph< V > Class Template Reference

Inheritance diagram for DefaultDirectedGraph< V >:



Protected Attributes

- std::set< V > vertices
- BinaryRelation< V, V > relation

4.2.1 Member Data Documentation

4.2.1.1 relation

```
template<typename V >
BinaryRelation<V, V> DefaultDirectedGraph< V >::relation  [protected]
```

4.2.1.2 vertices

```
template<typename V >
std::set<V> DefaultDirectedGraph< V >::vertices  [protected]
```

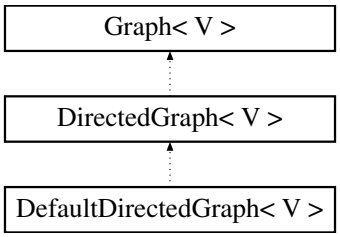
The documentation for this class was generated from the following file:

- src/graph/src/default_directed_graph.cpp

4.3 DirectedGraph< V > Class Template Reference

```
#include <directed_graph.h>
```

Inheritance diagram for DirectedGraph< V >:



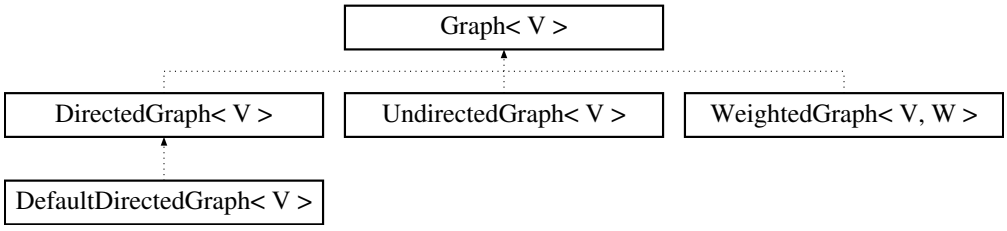
The documentation for this class was generated from the following file:

- src/graph/include/directed_graph.h

4.4 Graph< V > Class Template Reference

```
#include <graph.h>
```

Inheritance diagram for Graph< V >:



Public Member Functions

- virtual bool add_edge (V source, V target)
- virtual bool add_edge (Pair< V, V > pair)
- virtual bool add_vertex (V vertex)
- virtual bool contains_edge (Pair< V, V > edge)
- virtual bool contains_edge (V source, V target)
- virtual bool contains_vertex (V vertex)
- virtual std::set< Pair< V, V > > get_edge ()
- virtual std::set< Pair< V, V > > get_vertices ()
- virtual int get_number_of_vertices ()
- virtual int get_number_of_edges ()
- virtual bool set_number_of_vertices ()
- virtual bool set_number_of_edges ()
- virtual bool remove_vertex (V vertex)
- virtual bool remove_edge (std::set< Pair< V, V > >)
- virtual bool remove_edge (V source, V target)

4.4.1 Member Function Documentation

4.4.1.1 add_edge() [1/2]

```
template<typename V >
virtual bool Graph< V >::add_edge (
            V source,
            V target )  [virtual]
```

4.4.1.2 add_edge() [2/2]

```
template<typename V >
virtual bool Graph< V >::add_edge (
            Pair< V, V > pair )  [virtual]
```

4.4.1.3 add_vertex()

```
template<typename V >
virtual bool Graph< V >::add_vertex (
            V vertex )  [virtual]
```

4.4.1.4 contains_edge() [1/2]

```
template<typename V >
virtual bool Graph< V >::contains_edge (
            Pair< V, V > edge )  [virtual]
```

4.4.1.5 contains_edge() [2/2]

```
template<typename V >
virtual bool Graph< V >::contains_edge (
            V source,
            V target )  [virtual]
```

4.4.1.6 contains_vertex()

```
template<typename V >
virtual bool Graph< V >::contains_vertex (
            V vertex )  [virtual]
```

4.4.1.7 get_edge()

```
template<typename V >
virtual std::set<Pair<V, V> > Graph< V >::get_edge ( )  [virtual]
```

4.4.1.8 get_number_of_edges()

```
template<typename V >
virtual int Graph< V >::get_number_of_edges ( )  [virtual]
```

4.4.1.9 get_number_of_vertices()

```
template<typename V >
virtual int Graph< V >::get_number_of_vertices ( )  [virtual]
```

4.4.1.10 get_vertices()

```
template<typename V >
virtual std::set<Pair<V, V> > Graph< V >::get_vertices ( )  [virtual]
```

4.4.1.11 remove_edge() [1/2]

```
template<typename V >
virtual bool Graph< V >::remove_edge (
            std::set< Pair< V, V > >  )  [virtual]
```

4.4.1.12 remove_edge() [2/2]

```
template<typename V >
virtual bool Graph< V >::remove_edge (
            V source,
            V target )  [virtual]
```

4.4.1.13 remove_vertex()

```
template<typename V >
virtual bool Graph< V >::remove_vertex (
            V vertex )  [virtual]
```

4.4.1.14 set_number_of_edges()

```
template<typename V >
virtual bool Graph< V >::set_number_of_edges ( )  [virtual]
```

4.4.1.15 set_number_of_vertices()

```
template<typename V >
virtual bool Graph< V >::set_number_of_vertices ( )  [virtual]
```

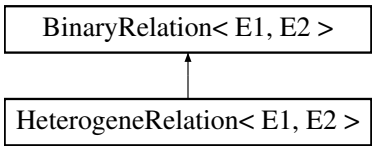
The documentation for this class was generated from the following file:

- src/graph/include/graph.h

4.5 HeterogeneRelation< E1, E2 > Class Template Reference

`#include <heterogene_relation.h>`

Inheritance diagram for HeterogeneRelation< E1, E2 >:



Public Member Functions

- HeterogeneRelation (const E1 &input_set, const E2 *target_set)
- bool isLeftTotal ()
- bool isRightTotal ()
- HeterogeneRelation< E1, E2 > get_allrelation (const HeterogeneRelation< E1, E2 > &relation)
- bool operator== (const HeterogeneRelation< E1, E2 > comp_rel)

Public Attributes

- bool isRightUnique
- bool isLeftUnique

4.5.1 Constructor & Destructor Documentation

4.5.1.1 HeterogeneRelation()

```
template<typename E1, typename E2>
HeterogeneRelation< E1, E2 >::HeterogeneRelation (
            const E1 & input_set,
            const E2 * target_set )  [inline]
```

4.5.2 Member Function Documentation

4.5.2.1 get_allrelation()

```
template<typename E1, typename E2>
HeterogeneRelation<E1, E2> HeterogeneRelation< E1, E2 >::get_allrelation (
            const HeterogeneRelation< E1, E2 > & relation )  [inline]
```

4.5.2.2 isLeftTotal()

```
template<typename E1, typename E2>
bool HeterogeneRelation< E1, E2 >::isLeftTotal ( )  [inline]
```

4.5.2.3 isRightTotal()

```
template<typename E1, typename E2>
bool HeterogeneRelation< E1, E2 >::isRightTotal ( )  [inline]
```

4.5.2.4 operator==()

```
template<typename E1, typename E2>
bool HeterogeneRelation< E1, E2 >::operator== (
            const HeterogeneRelation< E1, E2 > comp_rel )  [inline]
```

4.5.3 Member Data Documentation

4.5.3.1 isLeftUnique

```
template<typename E1, typename E2>
bool HeterogeneRelation< E1, E2 >::isLeftUnique
```

4.5.3.2 isRightUnique

```
template<typename E1, typename E2>
bool HeterogeneRelation< E1, E2 >::isRightUnique
```

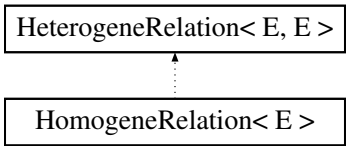
The documentation for this class was generated from the following file:

- src/relation/include/heterogene_relation.h

4.6 HomogeneRelation< E > Class Template Reference

`#include <homogene_relation.h>`

Inheritance diagram for HomogeneRelation< E >:



Public Member Functions

- bool isReflexive ()
- bool isSymmetric ()
- bool isTransitive ()
- HomogeneRelation< E > getReflexiveHull ()
- HomogeneRelation< E > getTransitiveHull ()

4.6.1 Member Function Documentation

4.6.1.1 getReflexiveHull()

```
template<typename E>
HomogeneRelation<E> HomogeneRelation< E >::getReflexiveHull ( )
```

4.6.1.2 getTransitiveHull()

```
template<typename E>
HomogeneRelation<E> HomogeneRelation< E >::getTransitiveHull ( )
```

4.6.1.3 isReflexive()

```
template<typename E>
bool HomogeneRelation< E >::isReflexive ( )
```

4.6.1.4 isSymmetric()

```
template<typename E>
bool HomogeneRelation< E >::isSymmetric ( )
```

4.6.1.5 isTransitive()

```
template<typename E>
bool HomogeneRelation< E >::isTransitive ( )
```

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

- src/relation/include/homogene_relation.h

4.7 Pair< E1, E2 > Class Template Reference

`#include <pair.h>`

Public Member Functions

- Pair (E1 e1, E2 e2)
- Pair (const Pair< E1, E2 > &pair)
- Pair< E2, E1 > transpose ()
- char * toString ()
- E1 getFirst ()
- E2 getSecond ()
- bool setFirst (E1 &e1)
- bool setSecond (E2 &e2)
- bool operator>= (const Pair< E1, E2 > &pair)
- bool operator<= (const Pair< E1, E2 > &pair)

4.7.1 Constructor & Destructor Documentation

4.7.1.1 Pair() [1/2]

```
template<typename E1, typename E2>
Pair< E1, E2 >::Pair (
            E1 e1,
            E2 e2 )  [inline]
```

4.7.1.2 Pair() [2/2]

```
template<typename E1, typename E2>
Pair< E1, E2 >::Pair (
            const Pair< E1, E2 > & pair )  [inline]
```

4.7.2 Member Function Documentation

4.7.2.1 getFirst()

```
template<typename E1, typename E2>
E1 Pair< E1, E2 >::getFirst ( )  [inline]
```

4.7.2.2 getSecond()

```
template<typename E1, typename E2>
E2 Pair< E1, E2 >::getSecond ( )  [inline]
```

4.7.2.3 operator<=()

```
template<typename E1, typename E2>
bool Pair< E1, E2 >::operator<= (
            const Pair< E1, E2 > & pair )  [inline]
```

4.7.2.4 operator>=()

```
template<typename E1, typename E2>
bool Pair< E1, E2 >::operator>= (
            const Pair< E1, E2 > & pair )  [inline]
```

4.7.2.5 setFirst()

```
template<typename E1, typename E2>
bool Pair< E1, E2 >::setFirst (
            E1 & e1 )  [inline]
```

4.7.2.6 setSecond()

```
template<typename E1, typename E2>
bool Pair< E1, E2 >::setSecond (
            E2 & e2 )  [inline]
```

4.7.2.7 toString()

```
template<typename E1, typename E2>
char* Pair< E1, E2 >::toString ( )  [inline]
```

4.7.2.8 transpose()

```
template<typename E1, typename E2>
Pair<E2, E1> Pair< E1, E2 >::transpose ( )  [inline]
```

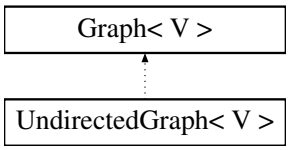
The documentation for this class was generated from the following file:

- src/relation/include/pair.h

4.8 UndirectedGraph< V > Class Template Reference

```
#include <undirected_graph.h>
```

Inheritance diagram for UndirectedGraph< V >:



Public Member Functions

- virtual int degree_Of (V vertex)
- virtual MySet< Pair< V, V > > get_neighbours (V vertex)
- virtual MySet< Pair< V, V > > neighbour_edges_of (V vertex)

4.8.1 Member Function Documentation

4.8.1.1 degree_Of()

```
template<typename V >
virtual int UndirectedGraph< V >::degree_Of (
            V vertex )  [virtual]
```

4.8.1.2 get_neighbours()

```
template<typename V >
virtual MySet<Pair<V, V> > UndirectedGraph< V >::get_neighbours (
            V vertex )  [virtual]
```

4.8.1.3 neighbour_edges_of()

```
template<typename V >
virtual MySet<Pair<V, V> > UndirectedGraph< V >::neighbour_edges_of (
            V vertex )  [virtual]
```

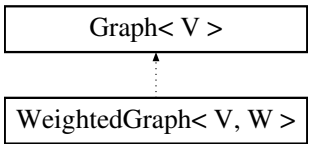
The documentation for this class was generated from the following file:

- src/graph/include/undirected_graph.h

4.9 WeightedGraph< V, W > Class Template Reference

```
#include <weighted_graph.h>
```

Inheritance diagram for WeightedGraph< V, W >:



Public Member Functions

- virtual bool add_edge (V source, V target, W weight)
- virtual bool add_edge (std::set< Pair< V, V >, W weight)
- virtual bool set_edge_weight (std::set< Pair< V, V >, W weight)
- virtual W get_edge_weight (std::set< Pair< V, V >)

4.9.1 Member Function Documentation

4.9.1.1 add_edge() [1/2]

```
template<typename V , typename W >
virtual bool WeightedGraph< V, W >::add_edge (
            V source,
            V target,
            W weight )  [virtual]
```

4.9.1.2 add_edge() [2/2]

```
template<typename V , typename W >
virtual bool WeightedGraph< V, W >::add_edge ( )  [virtual]
```

4.9.1.3 get_edge_weight()

```
template<typename V , typename W >
virtual W WeightedGraph< V, W >::get_edge_weight ( )  [virtual]
```

4.9.1.4 set_edge_weight()

```
template<typename V , typename W >
virtual bool WeightedGraph< V, W >::set_edge_weight ( )  [virtual]
```

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

- src/graph/include/weighted_graph.h

Chapter 5

File Documentation

5.1 src/graph/include/directed_graph.h File Reference

```
#include "graph.h"
#include <set>
```

Classes

- class DirectedGraph< V >

Macros

- #define DIR_GRAPH_N

5.1.1 Macro Definition Documentation

5.1.1.1 DIR_GRAPH_N

```
#define DIR_GRAPH_N
```

5.2 src/graph/include/graph.h File Reference

```
#include "pair.h"
#include <set>
```

Classes

- class Graph< V >

5.3 src/graph/include/undirected_graph.h File Reference

```
#include "graph.h"
```

Classes

- class UndirectedGraph< V >

5.4 src/graph/include/weighted_graph.h File Reference

```
#include "graph.h"
#include <set>
```

Classes

- class WeightedGraph< V, W >

5.5 src/graph/src/default_directed_graph.cpp File Reference

```
#include "directed_graph.h"
#include "binary_relation.h"
#include <set>
```

Classes

- class DefaultDirectedGraph< V >

5.6 src/relation/include/binary_relation.h File Reference

```
#include <set>
#include <stdio.h>
#include "pair.h"
```

Classes

- class BinaryRelation< E1, E2 >

5.6.1 Detailed Description

template class for a binary relation

Parameters

pair_set	A set of edges/ pairs for mapping a relation between two component
input_set	The input set of type <E1>, which contains all possible elements form that set
target_set	The target set of type <E1>, which contains all possible elements from that set

5.7 src/relation/include/heterogene_relation.h File Reference

```
#include "binary_relation.h"
#include <set>
```

Classes

- class HeterogeneRelation< E1, E2 >

Macros

- #define HET_REL_N

5.7.1 Macro Definition Documentation

5.7.1.1 HET_REL_N

```
#define HET_REL_N
```

5.8 src/relation/include/homogene_relation.h File Reference

```
#include <set>
#include "heterogene_relation.h"
```

Classes

- class HomogeneRelation< E >

Macros

- #define HOM_REL_N

5.8.1 Macro Definition Documentation

5.8.1.1 HOM_REL_N

```
#define HOM_REL_N
```

5.9 src/relation/include/pair.h File Reference

```
#include <iostream>
```

Classes

- class Pair< E1, E2 >

5.10 src/relation/src/main.cpp File Reference

```
#include "pair.h"
#include "binary_relation.h"
#include "heterogene_relation.h"
#include "homogene_relation.h"
```

Functions

- int main ()

5.10.1 Function Documentation

5.10.1.1 main()

```
int main ( )
```

Index