MMLT

Generated by Doxygen 1.8.1.2

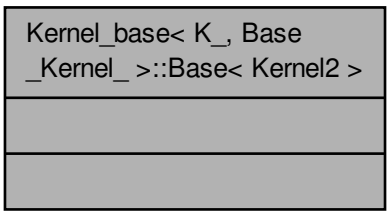
Tue Jun 25 2013 17:38:31

Contents

1 Class Documentation

1.1 Kernel_base< K_, Base_Kernel_ >::Base< Kernel2 > Struct Template Reference

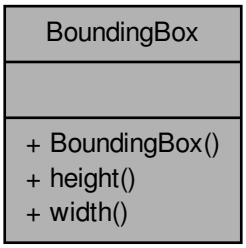
Collaboration diagram for Kernel_base< K_, Base_Kernel_ >::Base< Kernel2 >:



1.2 BoundingBox Class Reference

`#include <bounding_box.h>`

Collaboration diagram for BoundingBox:



Public Member Functions

- BoundingBox (const PointSet &points)
- Number height () const
- Number width () const

1.2.1 Constructor & Destructor Documentation

1.2.1.1 BoundingBox::BoundingBox (const PointSet & *points*) [inline]

1.2.2 Member Function Documentation

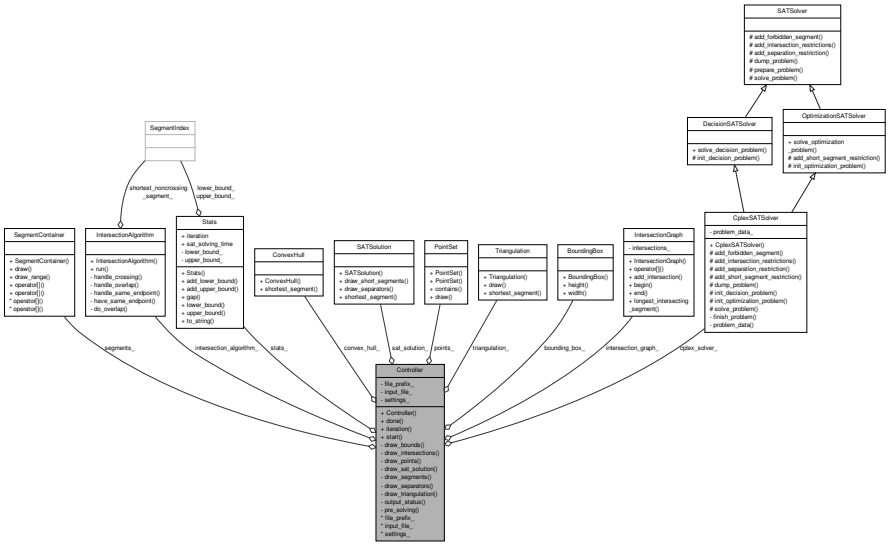
1.2.2.1 Number BoundingBox::height ( ) const [inline]

1.2.2.2 Number BoundingBox::width ( ) const [inline]

1.3 Controller Class Reference

```
#include <controller.h>
```

Collaboration diagram for Controller:



Public Member Functions

- Controller (const QString &file_prefix, QFile &input_file, const QSettings &settings)
- void done ()
- bool iteration ()
- bool start ()

Private Member Functions

- void draw_bounds () const
- void draw_intersections () const
- void draw_points (SVGPainter &painter) const
- void draw_sat_solution () const
- void draw_segments (SVGPainter &painter) const
- void draw_separators () const
- void draw_triangulation () const
- void output_status () const
- void pre_solving ()

Private Attributes

independent members

- CplexSATSolver cplex_solver_
- IntersectionAlgorithm intersection_algorithm_
- SATSolution sat_solution_
- Stats stats_

input parameters

- const QString & file_prefix_
- QFile & input_file_
- const QSettings & settings_

dependent on input parameter

- const PointSet points_

dependent on input points

- const BoundingBox bounding_box_
- const ConvexHull convex_hull_
- SegmentContainer segments_
- Triangulation triangulation_

dependent on segments

- IntersectionGraph intersection_graph_

1.3.1 Constructor & Destructor Documentation

1.3.1.1 Controller::Controller (const QString & *file_prefix,* QFile & *input_file,* const QSettings & *settings*)

1.3.2 Member Function Documentation

1.3.2.1 void Controller::done ( )

called after the algorithm finished

1.3.2.2 void Controller::draw_bounds ( ) const [private]

1.3.2.3 void Controller::draw_intersections ( ) const [private]

1.3.2.4 void Controller::draw_points (SVGPainter & *painter*) const [private]

1.3.2.5 void Controller::draw_sat_solution ( ) const [private]

1.3.2.6 void Controller::draw_segments (SVGPainter & *painter*) const [private]

1.3.2.7 void Controller::draw_separators ( ) const [private]

1.3.2.8 void Controller::draw_triangulation ( ) const [private]

1.3.2.9 bool Controller::iteration ( )

run next iteration

Returns

true if next iteration should be triggered

1.3.2.10 void Controller::output_status ( ) const [private]

dumps the current algorithm status

1.3.2.11 void Controller::pre_solving ( ) [private]

does some pre-processing

1.3.2.12 bool Controller::start ( )

start the algorithm

Returns

true if iteration should be triggered

1.3.3 Member Data Documentation

1.3.3.1 const BoundingBox Controller::bounding_box_ [private]

1.3.3.2 const ConvexHull Controller::convex_hull_ [private]

1.3.3.3 CplexSATSolver Controller::cplex_solver_ [private]

1.3.3.4 const QString& Controller::file_prefix_ [private]

1.3.3.5 QFile& Controller::input_file_ [private]

1.3.3.6 IntersectionAlgorithm Controller::intersection_algorithm_ [private]

1.3.3.7 IntersectionGraph Controller::intersection_graph_ [private]

1.3.3.8 const PointSet Controller::points_ [private]

1.3.3.9 SATSolution Controller::sat_solution_ [private]

1.3.3.10 SegmentContainer Controller::segments_ [private]

1.3.3.11 const QSettings& Controller::settings_ [private]

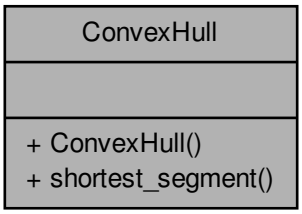
1.3.3.12 Stats Controller::stats_ [private]

1.3.3.13 Triangulation Controller::triangulation_ [private]

1.4 ConvexHull Class Reference

`#include <convex_hull.h>`

Collaboration diagram for ConvexHull:



Public Member Functions

- ConvexHull (const PointSet &points)
- const SegmentIndex & shortest_segment (const SegmentContainer &segments) const

1.4.1 Constructor & Destructor Documentation

1.4.1.1 ConvexHull::ConvexHull (const PointSet & *points*)

compute convex hull of given point set

1.4.2 Member Function Documentation

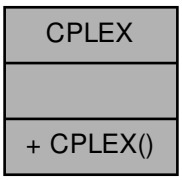
1.4.2.1 const SegmentIndex & ConvexHull::shortest_segment (const SegmentContainer & *segments*) const

find the convex hull segment with minimum length

1.5 CPLEX Class Reference

`#include <concert.h>`

Collaboration diagram for CPLEX:



Public Member Functions

- CPLEX ()

1.5.1 Detailed Description

ugly CPLEX code is not our fault helper class for CPLEX concert API

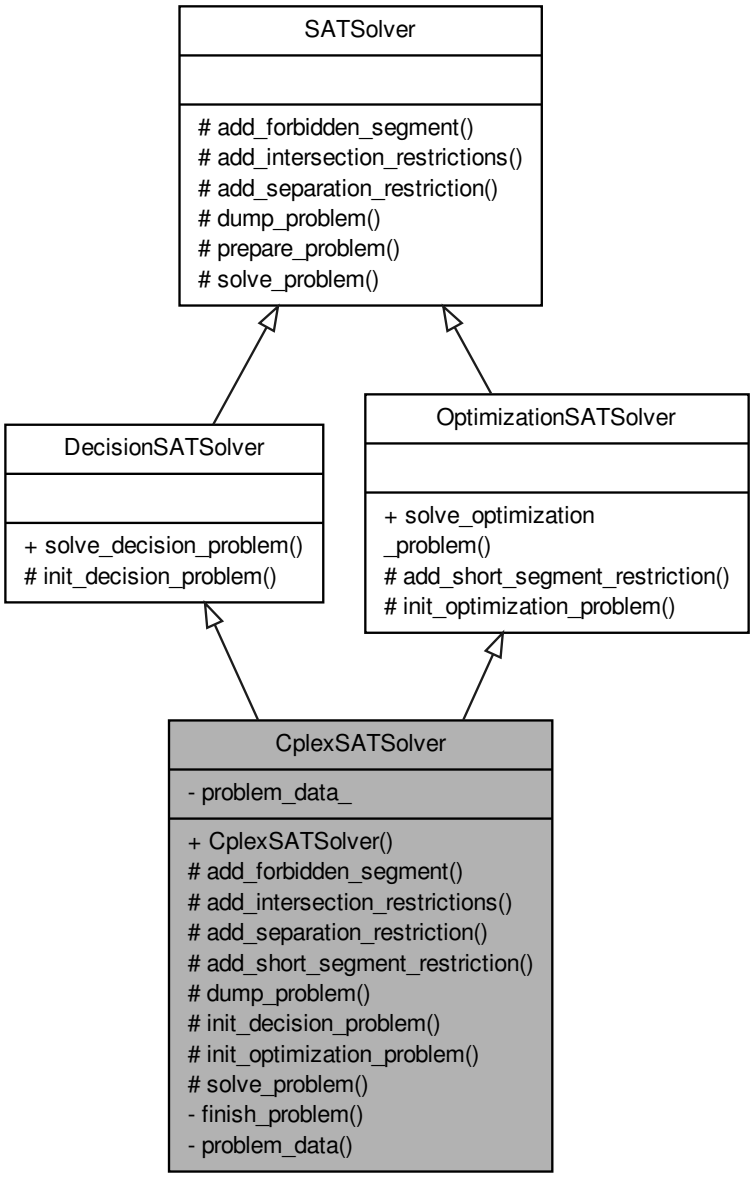
1.5.2 Constructor & Destructor Documentation

1.5.2.1 CPLEX::CPLEX ( ) [inline]

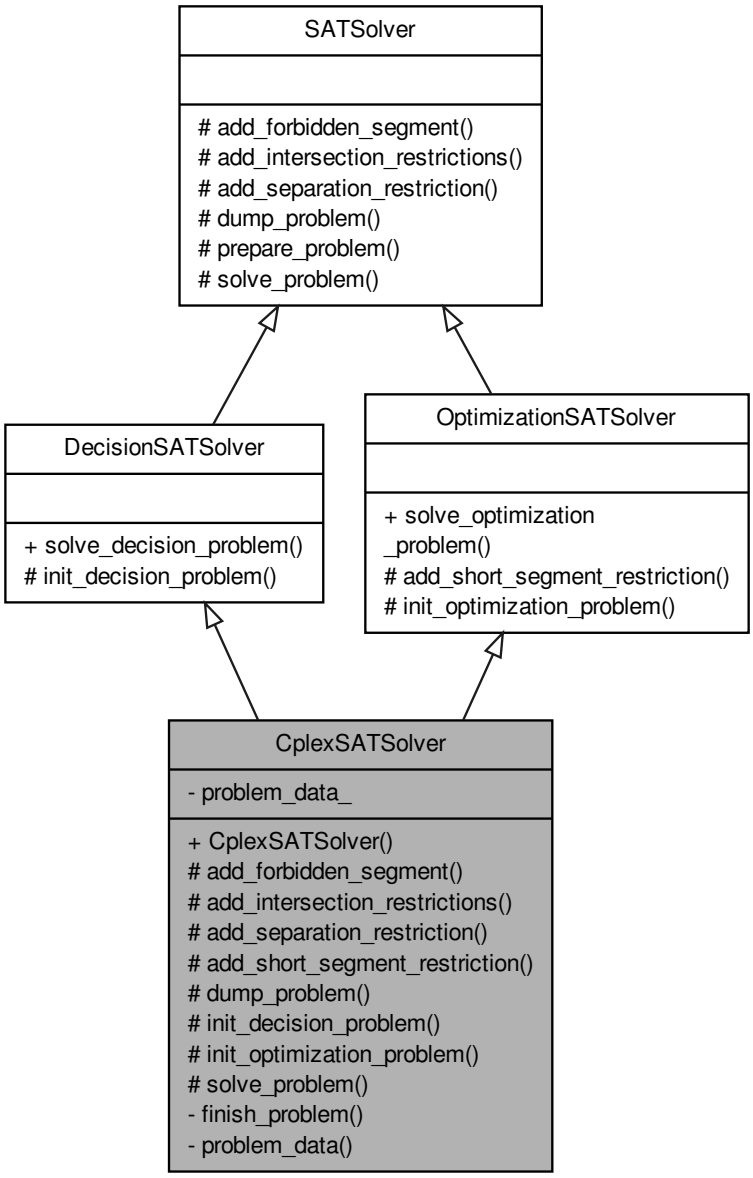
1.6 CplexSATSolver Class Reference

`#include <cplex_sat_solver.h>`

Inheritance diagram for CplexSATSolver:



Collaboration diagram for CplexSATSolver:



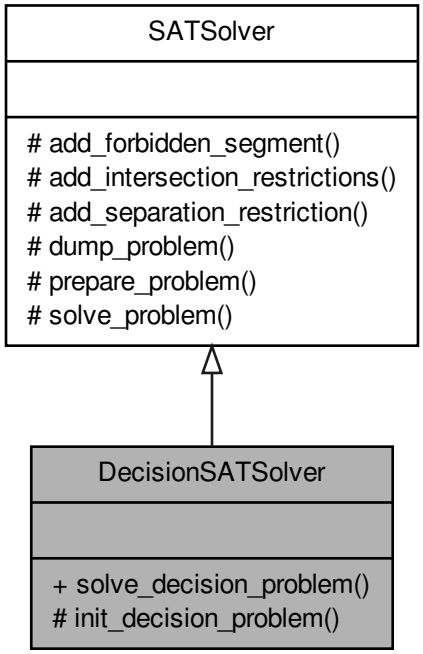
Classes

- struct ProblemData

Public Member Functions

- CplexSATSolver ()

Collaboration diagram for DecisionSATSolver:



Public Member Functions

- void solve_decision_problem (const QSettings &settings, const QString &file_prefix, const SATProblem &problem, SATSolution &solution)

Protected Member Functions

- virtual void init_decision_problem (const SATProblem ∗problem)=0

1.6.1 Detailed Description

interface for decision SAT solvers

1.6.2 Member Function Documentation

1.6.2.1 virtual void DecisionSATSolver::init_decision_problem (const SATProblem ∗ *problem*) [protected], [pure virtual]

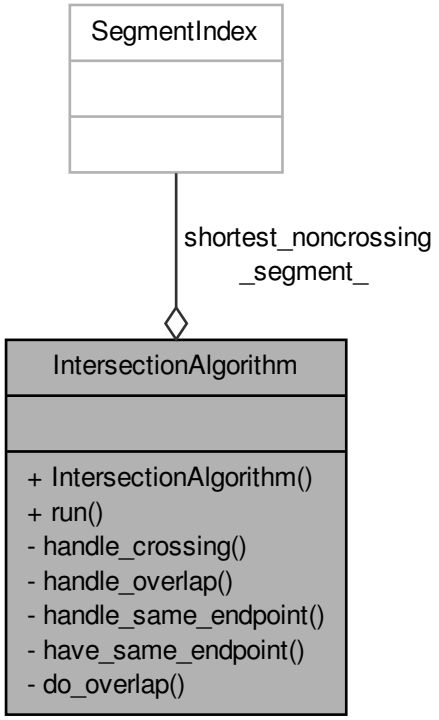
Implemented in CplexSATSolver.

1.6.2.2 void DecisionSATSolver::solve_decision_problem (const QSettings & *settings,* const QString & *file_prefix,* const SATProblem & *problem,* SATSolution & *solution*)

1.7 IntersectionAlgorithm Class Reference

`#include <intersection_algorithm.h>`

Collaboration diagram for IntersectionAlgorithm:



Public Member Functions

- IntersectionAlgorithm ()
- void run (IntersectionGraph &igraph, SegmentContainer &segments)

Public Attributes

- SegmentIndex shortest_noncrossing_segment_

Private Member Functions

- void handle_crossing (IntersectionGraph &igraph, const Segment &s1, const Segment &s2)
- void handle_overlap (IntersectionGraph &igraph, const Segment &s1, const Segment &s2)
- void handle_same_endpoint (const Segment &s1, const Segment &s2) const
- bool have_same_endpoint (const Segment &s1, const Segment &s2) const
- bool do_overlap (Segment &s1, Segment &s2) const

1.7.1 Constructor & Destructor Documentation

1.7.1.1 IntersectionAlgorithm::IntersectionAlgorithm ( )

1.7.2 Member Function Documentation

1.7.2.1 bool IntersectionAlgorithm::do_overlap (Segment & *s1,* Segment & *s2*) const [private]

checks if two segments overlap

Returns

the outer segment

1.7.2.2 void IntersectionAlgorithm::handle_crossing (IntersectionGraph & *igraph,* const Segment & *s1,* const Segment & *s2*) [private]

segments cross

1.7.2.3 void IntersectionAlgorithm::handle_overlap (IntersectionGraph & *igraph,* const Segment & *s1,* const Segment & *s2*) [private]

segments intersect but do not cross

1.7.2.4 void IntersectionAlgorithm::handle_same_endpoint (const Segment & *s1,* const Segment & *s2*) const [private]

segments have the same end point

1.7.2.5 bool IntersectionAlgorithm::have_same_endpoint (const Segment & *s1,* const Segment & *s2*) const [private]

checks if two segments share an endpoint

Returns

the endpoint

1.7.2.6 void IntersectionAlgorithm::run (IntersectionGraph & *igraph,* SegmentContainer & *segments*)

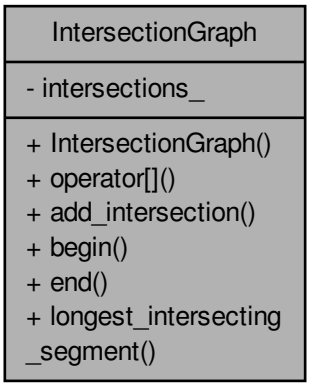
1.7.3 Member Data Documentation

1.7.3.1 SegmentIndex IntersectionAlgorithm::shortest_noncrossing_segment_

1.8 IntersectionGraph Class Reference

`#include <intersection_graph.h>`

Collaboration diagram for IntersectionGraph:



Public Member Functions

- IntersectionGraph (const SegmentIndex &size)
- const Intersections & operator[] (const SegmentIndex &index) const
- void add_intersection (const Segment &s1, const Segment &s2)
- IntersectionsVector::const_iterator begin () const
- IntersectionsVector::const_iterator end () const
- const SegmentIndex & longest_intersecting_segment (const SegmentIndex &index) const

Private Attributes

- IntersectionsVector intersections_

1.8.1 Constructor & Destructor Documentation

1.8.1.1 IntersectionGraph::IntersectionGraph (const SegmentIndex & *size*)

default constructor

1.8.2 Member Function Documentation

1.8.2.1 void IntersectionGraph::add_intersection (const Segment & *s1*, const Segment & *s2*)

add two intersecting segments to the graph

1.8.2.2 IntersectionsVector::const_iterator IntersectionGraph::begin ( ) const [inline]

1.8.2.3 IntersectionsVector::const_iterator IntersectionGraph::end ( ) const [inline]

1.8.2.4 const SegmentIndex & IntersectionGraph::longest_intersecting_segment (const SegmentIndex & *index*) const

1.8.2.5 const Intersections& IntersectionGraph::operator[] (const SegmentIndex & *index*) const [inline]

Returns

all intersecting segments for a segment

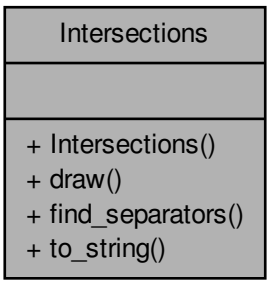
1.8.3 Member Data Documentation

1.8.3.1 IntersectionsVector IntersectionGraph::intersections_ [private]

1.9 Intersections Class Reference

`#include <intersections.h>`

Collaboration diagram for Intersections:



Public Member Functions

- Intersections ()
- void draw (QPainter &painter, const SegmentContainer &segments) const
- void find_separators (const SegmentIndex &segment_index, const SegmentContainer &segments, std::vector< SegmentIndex > &separators) const
- QString to_string (const SegmentContainer &segments) const

1.9.1 Detailed Description

sorted set of intersecting segments

1.9.2 Constructor & Destructor Documentation

1.9.2.1 Intersections::Intersections ( ) [inline]

default constructor

1.9.3 Member Function Documentation

1.9.3.1 void Intersections::draw (QPainter & *painter,* const SegmentContainer & *segments*) const

draws intersections using QPainter

1.9.3.2 void Intersections::find_separators (const SegmentIndex & *segment_index,* const SegmentContainer & *segments,* std::vector< SegmentIndex > & *separators*) const

finds all separators for a given segment and stores them in the passed container

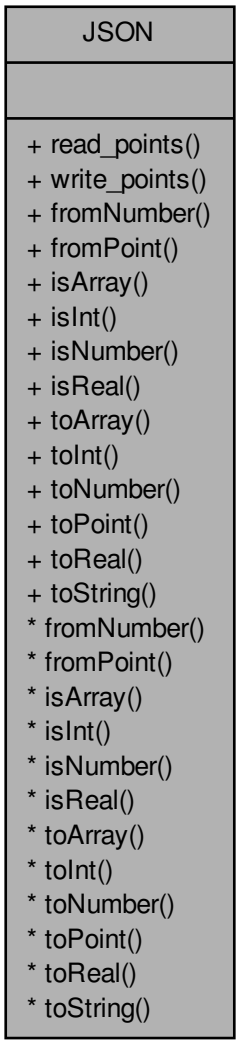
1.9.3.3 QString Intersections::to_string (const SegmentContainer & *segments*) const

output intersections to QString

1.10 JSON Class Reference

`#include <json.h>`

Collaboration diagram for JSON:



Static Public Member Functions

- template<typename OutputIterator >
  static bool read_points (QFile &file, OutputIterator output)
- template<typename Container >
  static bool write_points (const std::string &file_name, Container points)

helper functions

- static JSONValue fromNumber (const Number &value)
- static JSONArray fromPoint (const Point &point)
- static bool isArray (const JSONValue &value)
- static bool isInt (const JSONValue &value)
- static bool isNumber (const JSONValue &value)
- static bool isReal (const JSONValue &value)
- static const JSONArray & toArray (const JSONValue &value)
- static int toInt (const JSONValue &value)
- static Number toNumber (const JSONValue &value)
- static Point toPoint (const JSONValue &value)
- static double toReal (const JSONValue &value)
- static const std::string & toString (const JSONValue &value)

1.10.1 Member Function Documentation

1.10.1.1 JSON::JSONValue JSON::fromNumber (const Number & *value*) [static]

1.10.1.2 JSON::JSONArray JSON::fromPoint (const Point & *point*) [static]

1.10.1.3 bool JSON::isArray (const JSONValue & *value*) [static]

1.10.1.4 bool JSON::isInt (const JSONValue & *value*) [static]

1.10.1.5 bool JSON::isNumber (const JSONValue & *value*) [static]

1.10.1.6 bool JSON::isReal (const JSONValue & *value*) [static]

1.10.1.7 template<typename OutputIterator > static bool JSON::read_points (QFile & *file,* OutputIterator *output*) [inline], [static]

1.10.1.8 const JSON::JSONArray & JSON::toArray (const JSONValue & *value*) [static]

1.10.1.9 int JSON::toInt (const JSONValue & *value*) [static]

1.10.1.10 Number JSON::toNumber (const JSONValue & *value*) [static]

1.10.1.11 Point JSON::toPoint (const JSONValue & *value*) [static]

1.10.1.12 double JSON::toReal (const JSONValue & *value*) [static]

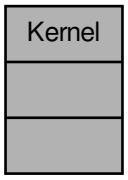
1.10.1.13 const std::string & JSON::toString (const JSONValue & *value*) [static]

1.10.1.14 template<typename Container > static bool JSON::write_points (const std::string & *file_name,* Container *points*) [inline], [static]

1.11 Kernel Struct Reference

`#include <kernel.h>`

Collaboration diagram for Kernel:



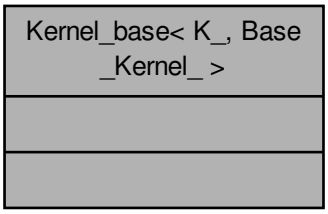
1.11.1 Detailed Description

customized kernel

1.12 Kernel_base< K_, Base_Kernel_ > Class Template Reference

`#include <kernel.h>`

Collaboration diagram for Kernel_base< K_, Base_Kernel_ >:



Classes

- struct Base

1.12.1 Detailed Description

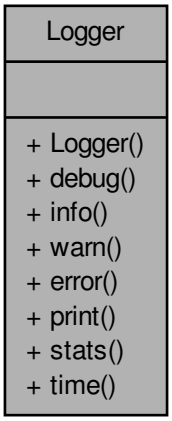
template<typename K_, typename Base_Kernel_>class Kernel_base< K_, Base_Kernel_ >

kernel base with customized PointC2 and SegmentC2

1.13 Logger Class Reference

`#include <logger.h>`

Collaboration diagram for Logger:



Public Member Functions

- Logger ()
- void debug (const QString &message) const
- void info (const QString &message) const
- void warn (const QString &message) const
- void error (const QString &message) const
- void print (const QString &message) const
- void stats (const Stats &stats) const
- void time (const QString &identifier, int milliseconds) const

1.13.1 Constructor & Destructor Documentation

1.13.1.1 Logger::Logger ( )

1.13.2 Member Function Documentation

1.13.2.1 void Logger::debug (const QString & *message*) const

1.13.2.2 void Logger::error (const QString & *message*) const

1.13.2.3 void Logger::info (const QString & *message*) const

1.13.2.4 void Logger::print (const QString & *message*) const

1.13.2.5 void Logger::stats (const Stats & *stats*) const

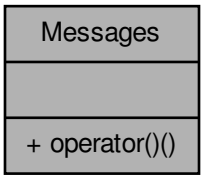
1.13.2.6 void Logger::time (const QString & *identifier,* int *milliseconds*) const

1.13.2.7 void Logger::warn (const QString & *message*) const

1.14 Messages Class Reference

`#include <logger.h>`

Collaboration diagram for Messages:



Public Member Functions

- QString operator() (const char *text)

1.14.1 Detailed Description

helper class for string literals

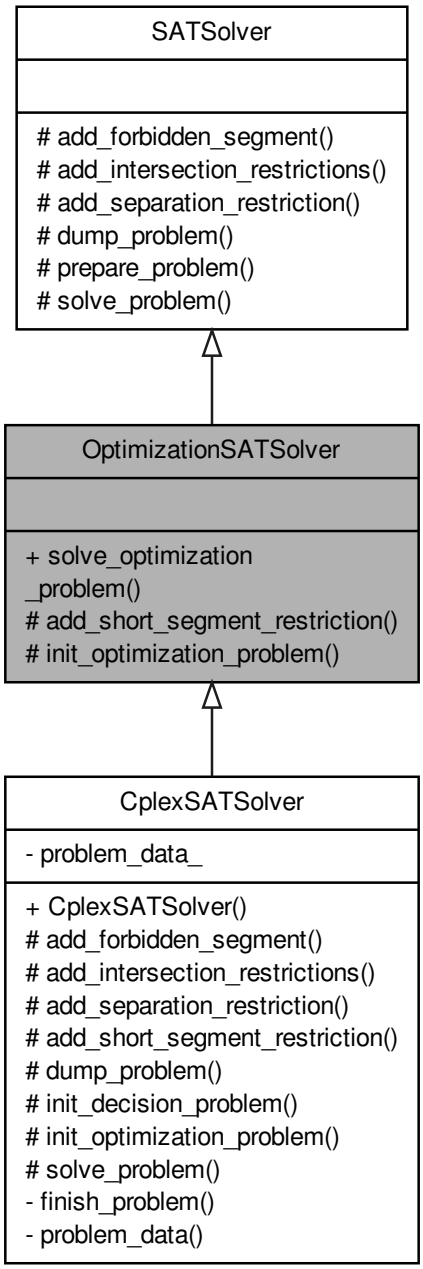
1.14.2 Member Function Documentation

1.14.2.1 QString Messages::operator() (const char * *text*) [inline]

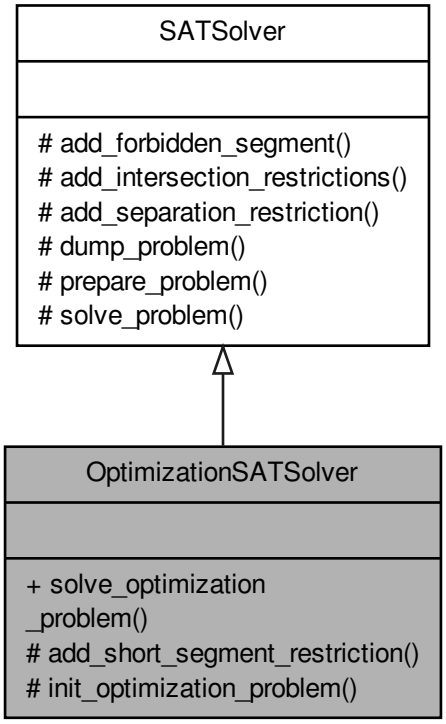
1.15 OptimizationSATSolver Class Reference

```
#include <sat_solver.h>
```

Inheritance diagram for OptimizationSATSolver:



Collaboration diagram for OptimizationSATSolver:



Public Member Functions

- void solve_optimization_problem (const QSettings &settings, const QString &file_prefix, const SATProblem &problem, SATSolution &solution)

Protected Member Functions

- virtual void add_short_segment_restriction (const SATProblem ∗problem, const SegmentIndex &index)=0
- virtual void init_optimization_problem (const SATProblem ∗problem)=0

1.15.1 Detailed Description

interface for optimization SAT solvers

1.15.2 Member Function Documentation

1.15.2.1 virtual void OptimizationSATSolver::add_short_segment_restriction (const SATProblem ∗ *problem,* const SegmentIndex & *index*) [protected], [pure virtual]

Implemented in CplexSATSolver.

1.15.2.2 virtual void OptimizationSATSolver::init_optimization_problem (const SATProblem * *problem*) [protected],[pure virtual]

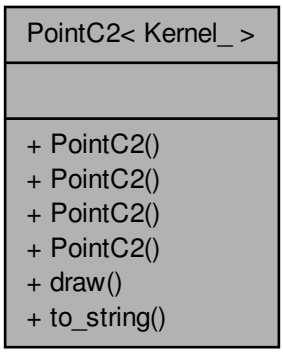
Implemented in CplexSATSolver.

1.15.2.3 void OptimizationSATSolver::solve_optimization_problem (const QSettings & *settings,* const QString & *file_prefix,* const SATProblem & *problem,* SATSolution & *solution*)

1.16 PointC2< Kernel_ > Class Template Reference

`#include <point.h>`

Collaboration diagram for PointC2< Kernel_ >:



Public Member Functions

- PointC2 ()
- PointC2 (const CGAL::Origin &origin)
- PointC2 (const FT &x, const FT &y)
- PointC2 (const FT &hx, const FT &hy, const FT &hw)
- void draw (QPainter &painter) const
- QString to_string () const

Private Types

- typedef Kernel_::FT FT
- typedef CGAL::PointC2< Kernel_ > PointBase

1.16.1 Detailed Description

template<class Kernel_>class PointC2< Kernel_ >

customized point type

1.16.2 Member Typedef Documentation

1.16.2.1 template<class Kernel_ > typedef Kernel_::FT PointC2< Kernel_ >::FT [private]

1.16.2.2 template<class Kernel_ > typedef CGAL::PointC2<Kernel_> PointC2< Kernel_ >::PointBase [private]

1.16.3 Constructor & Destructor Documentation

1.16.3.1 template<class Kernel_ > PointC2< Kernel_ >::PointC2 ( ) [inline]

empty constructor

1.16.3.2 template<class Kernel_ > PointC2< Kernel_ >::PointC2 (const CGAL::Origin & *origin*) [inline]

origin constructor

1.16.3.3 template<class Kernel_ > PointC2< Kernel_ >::PointC2 (const FT & *x,* const FT & *y*) [inline]

Cartesian constructor

1.16.3.4 template<class Kernel_ > PointC2< Kernel_ >::PointC2 (const FT & *hx,* const FT & *hy,* const FT & *hw*) [inline]

homogeneous constructor

1.16.4 Member Function Documentation

1.16.4.1 template<class Kernel_ > void PointC2< Kernel_ >::draw (QPainter & *painter*) const

draw segment using given QPainter

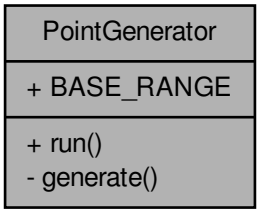
1.16.4.2 template<class Kernel_ > QString PointC2< Kernel_ >::to_string ( ) const

dump point to QString

1.17 PointGenerator Class Reference

`#include <point_generator.h>`

Collaboration diagram for PointGenerator:



Static Public Member Functions

- static void run (const QSettings &settings)

Static Public Attributes

- static const double BASE_RANGE = 100.0

Static Private Member Functions

- template<typename GeneratorType >
  static void generate (const QString &base_name, std::size_t num_points, std::size_t num_iterations)

1.17.1 Member Function Documentation

1.17.1.1 template<typename GeneratorType > static void PointGenerator::generate (const QString & *base_name,* std::size_t *num_points,* std::size_t *num_iterations*) [inline], [static], [private]

1.17.1.2 static void PointGenerator::run (const QSettings & *settings*) [inline], [static]

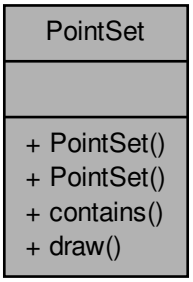
1.17.2 Member Data Documentation

1.17.2.1 const double PointGenerator::BASE_RANGE = 100.0 [static]

1.18 PointSet Class Reference

`#include <point_set.h>`

Collaboration diagram for PointSet:



Public Member Functions

- PointSet ()
- PointSet (QFile &input_file)
- bool contains (const Point &point) const
- void draw (QPainter &painter) const

1.18.1 Detailed Description

(sorted) set of points

1.18.2 Constructor & Destructor Documentation

1.18.2.1 PointSet::PointSet ( )

empty set

1.18.2.2 PointSet::PointSet (QFile & *input_file*)

read points from file

1.18.3 Member Function Documentation

1.18.3.1 bool PointSet::contains (const Point & *point*) const [inline]

shortcut for STL count()

Returns

true if point is in set

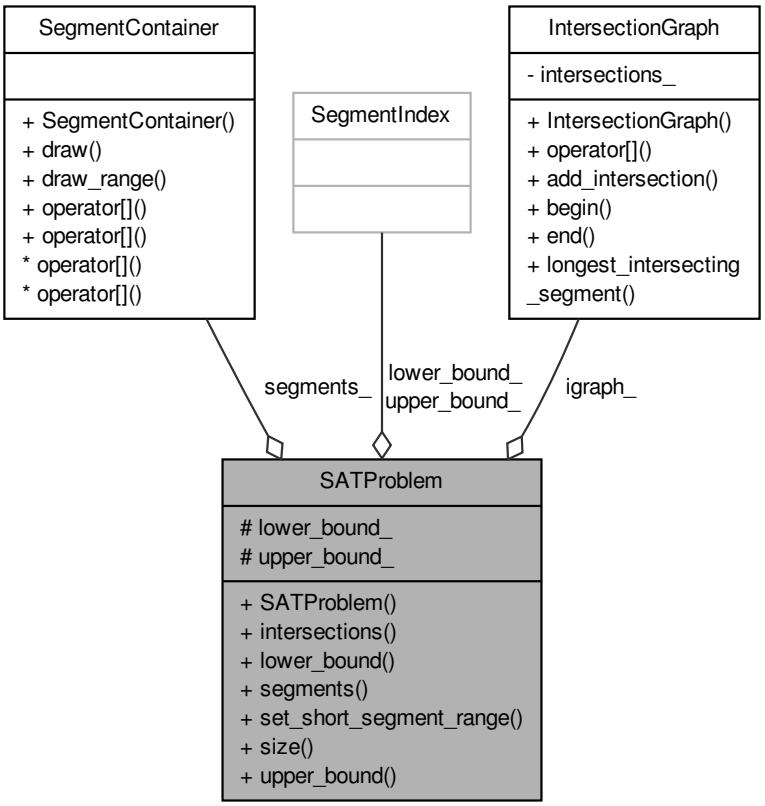
1.18.3.2 void PointSet::draw (QPainter & *painter*) const

output point set using QPainter

1.19 SATProblem Class Reference

`#include <sat_problem.h>`

Collaboration diagram for SATProblem:



Public Member Functions

- SATProblem (const IntersectionGraph &igraph, const SegmentContainer &segments)
- const Intersections & intersections (const SegmentIndex &index) const
- const SegmentIndex & lower_bound () const
- const SegmentContainer & segments () const
- void set_short_segment_range (const SegmentIndex &lower_bound, const SegmentIndex &upper_bound)
- SegmentIndex size () const
- const SegmentIndex & upper_bound () const

Protected Attributes

- const IntersectionGraph & igraph_
- const SegmentContainer & segments_
- SegmentIndex lower_bound_
- SegmentIndex upper_bound_

1.19.1 Constructor & Destructor Documentation

1.19.1.1 SATProblem::SATProblem (const IntersectionGraph & *igraph,* const SegmentContainer & *segments*)

default constructor

1.19.2 Member Function Documentation

1.19.2.1 const Intersections& SATProblem::intersections (const SegmentIndex & *index*) const `[inline]`

1.19.2.2 const SegmentIndex& SATProblem::lower_bound ( ) const `[inline]`

1.19.2.3 const SegmentContainer& SATProblem::segments ( ) const `[inline]`

1.19.2.4 void SATProblem::set_short_segment_range (const SegmentIndex & *lower_bound,* const SegmentIndex & *upper_bound*)

set range of segments to consider

1.19.2.5 SegmentIndex SATProblem::size ( ) const `[inline]`

1.19.2.6 const SegmentIndex& SATProblem::upper_bound ( ) const `[inline]`

1.19.3 Member Data Documentation

1.19.3.1 const IntersectionGraph& SATProblem::igraph_ `[protected]`

1.19.3.2 SegmentIndex SATProblem::lower_bound_ `[protected]`

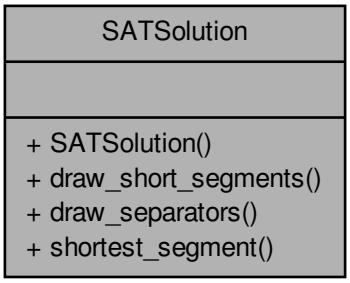
1.19.3.3 const SegmentContainer& SATProblem::segments_ `[protected]`

1.19.3.4 SegmentIndex SATProblem::upper_bound_ `[protected]`

1.20 SATSolution Class Reference

`#include <sat_solution.h>`

Collaboration diagram for SATSolution:



Public Member Functions

- SATSolution ()
- void draw_short_segments (QPainter &painter, const SegmentIndex &num_short_segments, const SegmentContainer &segments) const
- void draw_separators (QPainter &painter, const SegmentIndex &num_short_segments, const SegmentContainer &segments) const
- const SegmentIndex & shortest_segment () const

1.20.1 Constructor & Destructor Documentation

1.20.1.1 SATSolution::SATSolution ( ) [inline]

1.20.2 Member Function Documentation

1.20.2.1 void SATSolution::draw_separators (QPainter & *painter,* const SegmentIndex & *num_short_segments,* const SegmentContainer & *segments*) const

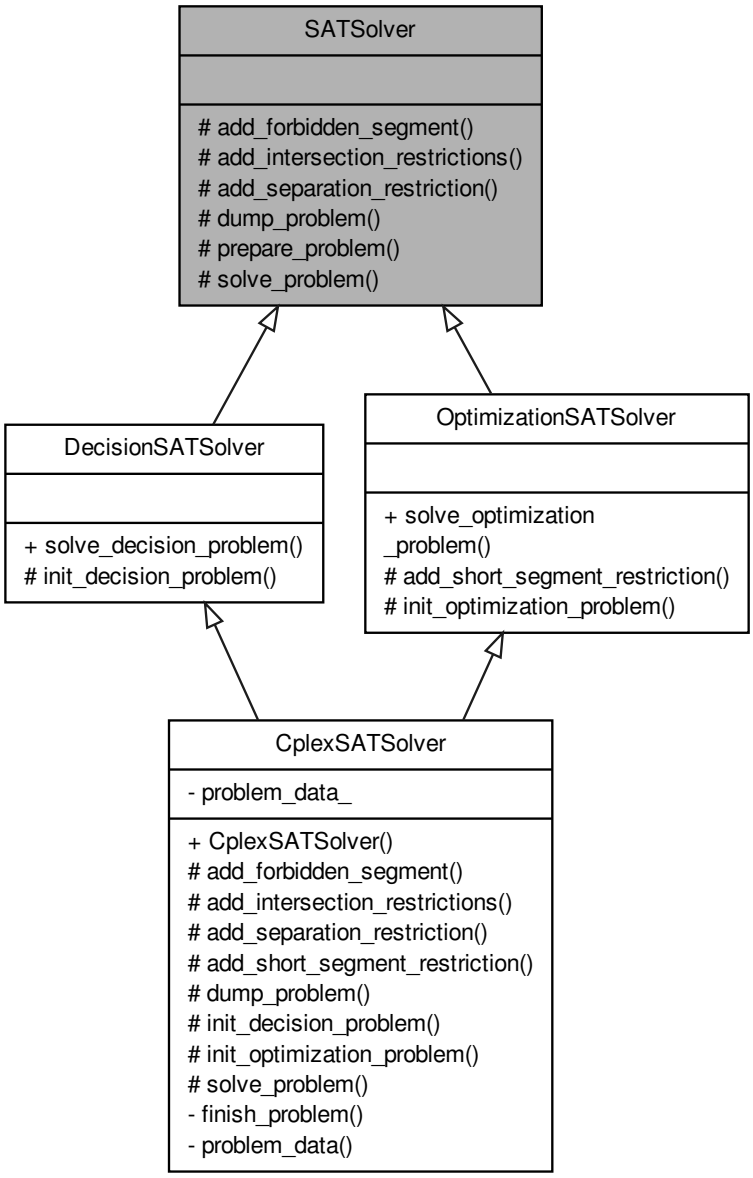
1.20.2.2 void SATSolution::draw_short_segments (QPainter & *painter,* const SegmentIndex & *num_short_segments,* const SegmentContainer & *segments*) const

1.20.2.3 const SegmentIndex & SATSolution::shortest_segment ( ) const

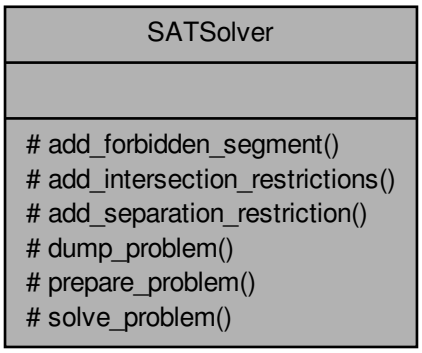
1.21 SATSolver Class Reference

`#include <sat_solver.h>`

Inheritance diagram for SATSolver:



Collaboration diagram for SATSolver:



Protected Member Functions

- virtual void add_forbidden_segment (const SATProblem *problem, const SegmentIndex &index)=0
- virtual void add_intersection_restrictions (const SATProblem *problem, const SegmentIndex &index, const Intersections &igroup)=0
- virtual void add_separation_restriction (const SATProblem *problem, const SegmentIndex &index, const std::vector< SegmentIndex > &separators)=0
- virtual void dump_problem (const QString &file_prefix, const SATProblem *problem)=0
- void prepare_problem (const SATProblem &problem)
- virtual void solve_problem (const SATProblem *problem, SATSolution &solution)=0

1.21.1 Detailed Description

interface for SAT solvers

1.21.2 Member Function Documentation

1.21.2.1 virtual void SATSolver::add_forbidden_segment (const SATProblem * *problem*, const SegmentIndex & *index*) [protected], [pure virtual]

Implemented in CplexSATSolver.

1.21.2.2 virtual void SATSolver::add_intersection_restrictions (const SATProblem * *problem*, const SegmentIndex & *index*, const Intersections & *igroup*) [protected], [pure virtual]

Implemented in CplexSATSolver.

1.21.2.3 virtual void SATSolver::add_separation_restriction (const SATProblem * *problem*, const SegmentIndex & *index*, const std::vector< SegmentIndex > & *separators*) [protected], [pure virtual]

Implemented in CplexSATSolver.

1.21.2.4 virtual void SATSolver::dump_problem (const QString & *file_prefix*, const SATProblem * *problem*) [protected], [pure virtual]

Implemented in CplexSATSolver.

1.21.2.5 void SATSolver::prepare_problem (const SATProblem & *problem*) [protected]

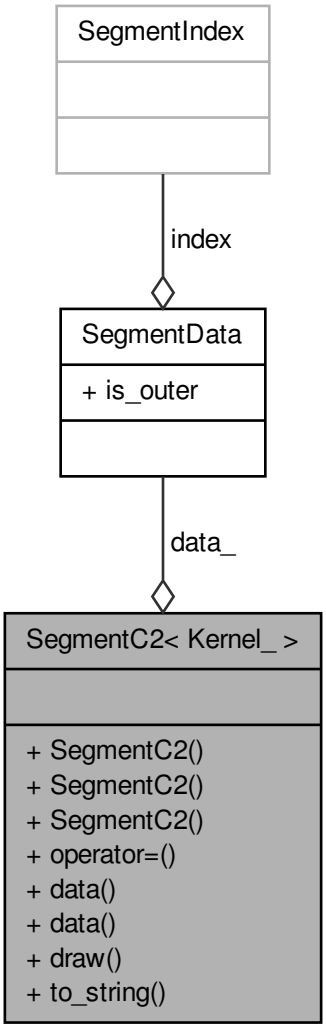
1.21.2.6 virtual void SATSolver::solve_problem (const SATProblem * *problem,* SATSolution & *solution*) [protected],[pure virtual]

Implemented in CplexSATSolver.

1.22 SegmentC2< Kernel_ > Class Template Reference

`#include <segment.h>`

Collaboration diagram for SegmentC2< Kernel_ >:



Public Member Functions

- SegmentC2 ()
- SegmentC2 (const Point_2 &source, const Point_2 &target)

- SegmentC2 (const SegmentC2 &other)
- SegmentC2 & operator= (const SegmentC2 &other)
- SegmentData & data ()
- const SegmentData & data () const
- void draw (QPainter &painter) const
- QString to_string () const

Private Attributes

- SegmentData data_

1.22.1 Detailed Description

template<class Kernel_>class SegmentC2< Kernel_ >

customized segment type

1.22.2 Constructor & Destructor Documentation

1.22.2.1 template<class Kernel_ > SegmentC2< Kernel_ >::SegmentC2 ( ) [inline]

empty constructor

1.22.2.2 template<class Kernel_ > SegmentC2< Kernel_ >::SegmentC2 (const Point_2 & *source,* const Point_2 & *target*) [inline]

base constructor

1.22.2.3 template<class Kernel_ > SegmentC2< Kernel_ >::SegmentC2 (const SegmentC2< Kernel_ > & *other*) [inline]

copy constructor

1.22.3 Member Function Documentation

1.22.3.1 template<class Kernel_ > SegmentData& SegmentC2< Kernel_ >::data ( ) [inline]

getter for attached data

1.22.3.2 template<class Kernel_ > const SegmentData& SegmentC2< Kernel_ >::data ( ) const [inline]

constant getter for attached data

1.22.3.3 template<class Kernel_ > void SegmentC2< Kernel_ >::draw (QPainter & *painter*) const

draw segment using given QPainter

1.22.3.4 template<class Kernel_ > SegmentC2& SegmentC2< Kernel_ >::operator= (const SegmentC2< Kernel_ > & *other*) [inline]

assignment operator

1.22.3.5 template<class Kernel_ > QString SegmentC2< Kernel_ >::to_string ( ) const

dump segment to QString

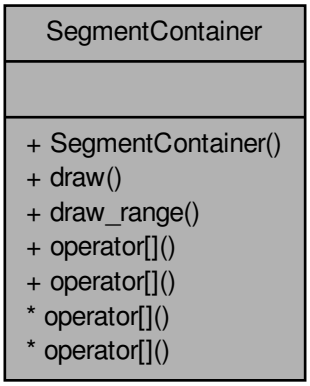
1.22.4 Member Data Documentation

1.22.4.1 template<class Kernel_ > SegmentData SegmentC2< Kernel_ >::data_ [private]

1.23 SegmentContainer Class Reference

`#include <segment_container.h>`

Collaboration diagram for SegmentContainer:



Public Member Functions

- SegmentContainer (const PointSet &points)
- void draw (QPainter &painter) const
- void draw_range (QPainter &painter, const SegmentIndex &lower_bound, const SegmentIndex &upper_bound) const

access i-th shortest segment

these operators assume that the segment set is not changed after construction

- Segment & operator[] (const SegmentIndex &index)
- const Segment & operator[] (const SegmentIndex &index) const

1.23.1 Detailed Description

container of segments sorted by length

1.23.2 Constructor & Destructor Documentation

1.23.2.1 SegmentContainer::SegmentContainer (const PointSet & *points*)

construct segments for all point pairs from set

1.23.3 Member Function Documentation

1.23.3.1 void SegmentContainer::draw (QPainter & *painter*) const

draws all segments

1.23.3.2 void SegmentContainer::draw_range (QPainter & *painter,* const SegmentIndex & *lower_bound,* const SegmentIndex & *upper_bound*) const

draw a range of segments

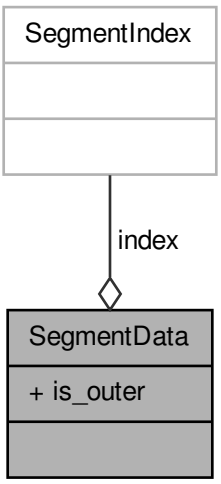
1.23.3.3 Segment & SegmentContainer::operator[] (const SegmentIndex & *index*)

1.23.3.4 const Segment & SegmentContainer::operator[] (const SegmentIndex & *index*) const

1.24 SegmentData Struct Reference

`#include <segment.h>`

Collaboration diagram for SegmentData:



Public Attributes

- SegmentIndex index
- bool is_outer

1.24.1 Detailed Description

data attached to a segment

1.24.2 Member Data Documentation

1.24.2.1 SegmentIndex SegmentData::index

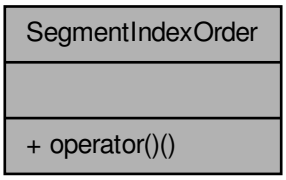
1.24.2.2 bool SegmentData::is_outer

true if the segment includes another

1.25 SegmentIndexOrder Struct Reference

`#include <orders.h>`

Collaboration diagram for SegmentIndexOrder:



Public Member Functions

- CGAL::Comparison_result operator() (const Segment &s, const Segment &t) const

1.25.1 Detailed Description

CGAL order for Segment by index

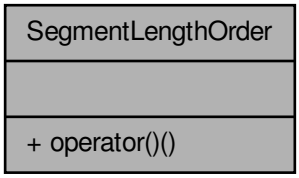
1.25.2 Member Function Documentation

1.25.2.1 CGAL::Comparison_result SegmentIndexOrder::operator() (const Segment & *s*, const Segment & *t*) const

1.26 SegmentLengthOrder Struct Reference

`#include <orders.h>`

Collaboration diagram for SegmentLengthOrder:



Public Member Functions

- CGAL::Comparison_result operator() (const Segment &s, const Segment &t) const

1.26.1 Detailed Description

CGAL order for Segment by length

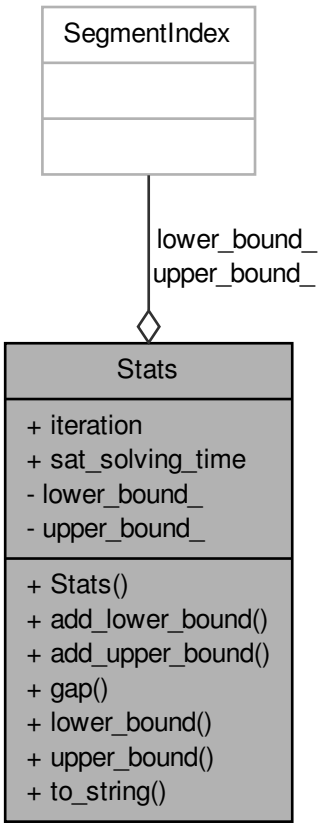
1.26.2 Member Function Documentation

1.26.2.1 CGAL::Comparison_result SegmentLengthOrder::operator() (const Segment & *s*, const Segment & *t*) const

1.27 Stats Class Reference

`#include <stats.h>`

Collaboration diagram for Stats:



Public Member Functions

- Stats ()
- void add_lower_bound (const SegmentIndex &bound)

- void add_upper_bound (const SegmentIndex &bound)
- SegmentIndex gap () const
- const SegmentIndex & lower_bound () const
- const SegmentIndex & upper_bound () const
- QString to_string () const

Public Attributes

- size_t iteration
- quint64 sat_solving_time

Private Attributes

- SegmentIndex lower_bound_
- SegmentIndex upper_bound_

1.27.1 Constructor & Destructor Documentation

1.27.1.1 Stats::Stats ( ) [inline]

1.27.2 Member Function Documentation

1.27.2.1 void Stats::add_lower_bound (const SegmentIndex & *bound*) [inline]

1.27.2.2 void Stats::add_upper_bound (const SegmentIndex & *bound*) [inline]

1.27.2.3 SegmentIndex Stats::gap ( ) const [inline]

1.27.2.4 const SegmentIndex& Stats::lower_bound ( ) const [inline]

1.27.2.5 QString Stats::to_string ( ) const [inline]

1.27.2.6 const SegmentIndex& Stats::upper_bound ( ) const [inline]

1.27.3 Member Data Documentation

1.27.3.1 size_t Stats::iteration

1.27.3.2 SegmentIndex Stats::lower_bound_ [private]

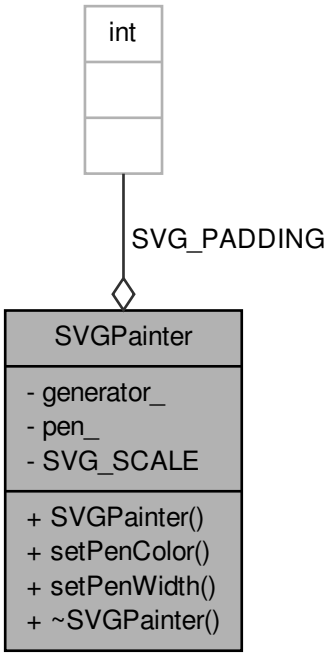
1.27.3.3 quint64 Stats::sat_solving_time

1.27.3.4 SegmentIndex Stats::upper_bound_ [private]

1.28 SVGPainter Class Reference

`#include <svg_painter.h>`

Collaboration diagram for SVGPainter:



Public Member Functions

- SVGPainter (const QString &file_prefix, const QString &file_name, const BoundingBox &bbox)
- void setPenColor (const QColor &color)
- void setPenWidth (int width)
- ~SVGPainter ()

Private Attributes

- QSvgGenerator generator_
- QPen pen_

Static Private Attributes

- static const int SVG_PADDING = 10
- static const double SVG_SCALE = 4.0

1.28.1 Constructor & Destructor Documentation

1.28.1.1 SVGPainter::SVGPainter (const QString & *file_prefix,* const QString & *file_name,* const BoundingBox & *bbox*)

1.28.1.2 SVGPainter::~SVGPainter ( )

1.28.2 Member Function Documentation

1.28.2.1 void SVGPainter::setPenColor (const QColor & *color*)

1.28.2.2 void SVGPainter::setPenWidth (int *width*)

1.28.3 Member Data Documentation

1.28.3.1 QSvgGenerator SVGPainter::generator_ [private]

1.28.3.2 QPen SVGPainter::pen_ [private]

1.28.3.3 const int SVGPainter::SVG_PADDING = 10 [static], [private]

padding for SVG images

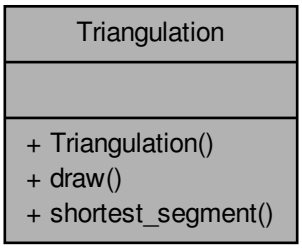
1.28.3.4 const double SVGPainter::SVG_SCALE = 4.0 [static], [private]

scale for SVG images

1.29 Triangulation Class Reference

`#include <triangulation.h>`

Collaboration diagram for Triangulation:



Public Member Functions

- Triangulation (const PointSet &points)
- void draw (QPainter &painter) const
- const SegmentIndex & shortest_segment (const SegmentContainer &segments) const

1.29.1 Constructor & Destructor Documentation

1.29.1.1 Triangulation::Triangulation (const PointSet & *points*)

default constructor

1.29.2 Member Function Documentation

1.29.2.1 void Triangulation::draw (QPainter & *painter*) const

draw triangulation segments using given QPainter

1.29.2.2 const SegmentIndex & Triangulation::shortest_segment (const SegmentContainer & *segments*) const