GPUSAT

Generated by Doxygen 1.8.18

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

Namespace Documentation

4.1 gpusat Namespace Reference

Classes

- class BagMatrix
- struct bagType

    type for a bag in the tree decomposition
- class CNFParser
- class CutSetWidthFitnessFunction
- class Decomposer
- class Graphoutput
- class JoinSizeFitnessFunction
- class NumJoinFitnessFunction
- class Preprocessor
- struct satformulaType

    type for saving the sat formula
- class Solver
- struct TableLines
- class TDParser
- struct treedecType

    type for saving a tree decomposition
- struct treeType

    tree type for storing the models
- class Visualization
- class WidthCutSetFitnessFunction

Enumerations

- enum dataStructure { dataStructure::ARRAY, dataStructure::TREE }
- enum nodeType { nodeType::JOIN, nodeType::INTRODUCEFORGET }

Functions

- bool compTreedType (const bagType ∗a, const bagType ∗b)
- bool compVars (const cl_long &a, const cl_long &b)
- cl_double getCount (cl_long id, cl_long ∗tree, cl_long numVars)
- long long int getTime ()
- std::ostream & operator<< (std::ostream &os, const dataStructure ds)
- std::ostream & operator<< (std::ostream &os, const std::vector< bagType ∗ > vec)
- std::ostream & operator<< (std::ostream &os, const std::vector< cl_long > vec)
- void printbagType (bagType ∗bag, std::ostream &stream, int depth=0)

  print information for a bag in the tree decomposition
- void printtreedecType (treedecType ∗dec, std::ostream &stream)

  print a tree decomposition
- void printtreeType (treeType ∗tree, std::ostream &stream, size_t size, int depth=0)
- TableLines solJson (bagType node, dataStructure solutionType)
- std::string solutiontable (bagType node, dataStructure solutionType)

  Generate a formatted stringoutput for a solved node with solutions.

4.1.1 Enumeration Type Documentation

4.1.1.1 dataStructure

```
enum gpusat::dataStructure  [strong]
```

Enumerator

ARRAY	
TREE	

4.1.1.2 nodeType

```
enum gpusat::nodeType  [strong]
```

Enumerator

JOIN	
INTRODUCEFORGET	

4.1.2 Function Documentation

4.1.2.1 compTreedType()

```
bool gpusat::compTreedType (
             const bagType * a,
             const bagType * b )  [inline]
```

Function that compares two tree decompostions by id.

Parameters

a	the first tree decompostion
b	the second tree decomposition

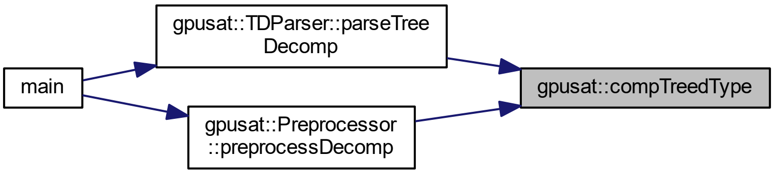
Returns

a < b

References gpusat::bagType::id.

Referenced by gpusat::TDParser::parseTreeDecomp(), and gpusat::Preprocessor::preprocessDecomp().

Here is the caller graph for this function:


main
gpusat::TDParser::parseTree Decomp
gpusat::Preprocessor ::preprocessDecomp
gpusat::compTreedType

4.1.2.2 compVars()

```
bool gpusat::compVars (
             const cl_long & a,
             const cl_long & b )  [inline]
```

Function that compares two variables.

Parameters

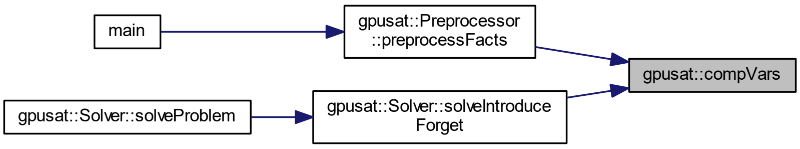
a	the first variable
b	the second variable

Returns

true if abs a < b

Referenced by gpusat::Preprocessor::preprocessFacts(), and gpusat::Solver::solveIntroduceForget().

Here is the caller graph for this function:


main
gpusat::Preprocessor ::preprocessFacts
gpusat::compVars
gpusat::Solver::solveProblem
gpusat::Solver::solveIntroduce Forget

4.1.2.3 getCount()

```
cl_double gpusat::getCount (
            cl_long id,
            cl_long * tree,
            cl_long numVars )  [inline]
```

returns the model count which corresponds to the given id

Parameters

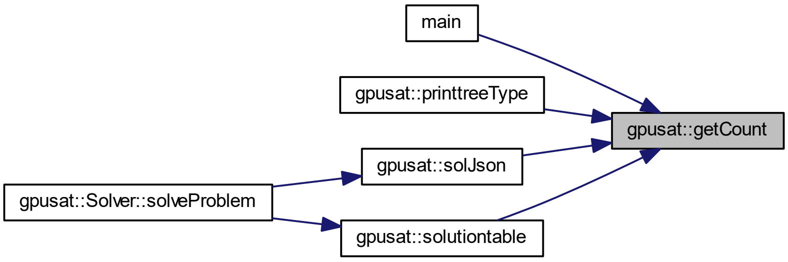
id	the id for which the model count should be returned
tree	a pointer to the tree structure
numVars	the number of variables in the bag

Returns

the model count

Referenced by main(), printtreeType(), solJson(), and solutiontable().

Here is the caller graph for this function:



4.1.2.4 getTime()

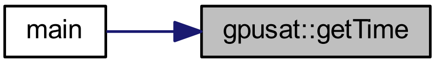
```
long long int gpusat::getTime ( )  [inline]
```

Returns

the time in millisecons since the epoch

Referenced by main().

Here is the caller graph for this function:



4.1.2.5 operator<<() [1/3]

```
std::ostream& gpusat::operator<< (
             std::ostream & os,
             const dataStructure ds )  [inline]
```

References ARRAY, and TREE.

4.1.2.6 operator<<() [2/3]

```
std::ostream& gpusat::operator<< (
            std::ostream & os,
            const std::vector< bagType * > vec )  [inline]
```

4.1.2.7 operator<<() [3/3]

```
std::ostream& gpusat::operator<< (
            std::ostream & os,
            const std::vector< cl_long > vec )  [inline]
```

4.1.2.8 printbagType()

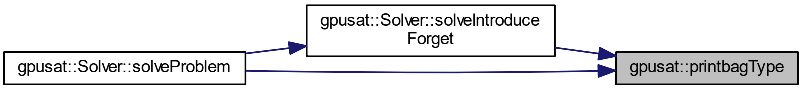
```
void gpusat::printbagType (
            bagType * bag,
            std::ostream & stream,
            int depth = 0 )  [inline]
```

print information for a bag in the tree decomposition

References gpusat::bagType::bags, gpusat::bagType::correction, gpusat::bagType::edges, gpusat::bagType::exponent, gpusat::bagType::id, and gpusat::bagType::variables.

Referenced by gpusat::Solver::solveIntroduceForget(), and gpusat::Solver::solveProblem().

Here is the caller graph for this function:



4.1.2.9 printtreedecType()

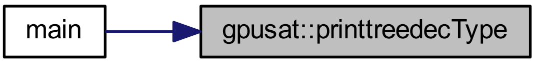
```
void gpusat::printtreedecType (
            treedecType * dec,
            std::ostream & stream )  [inline]
```

print a tree decomposition

References gpusat::treedecType::bags, and gpusat::treedecType::numb.

Referenced by main().

Here is the caller graph for this function:

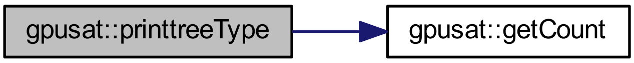


4.1.2.10 printtreeType()

```
void gpusat::printtreeType (
            treeType * tree,
            std::ostream & stream,
            size_t size,
            int depth = 0 )  [inline]
```

References gpusat::treeType::elements, getCount(), gpusat::treeType::maxId, gpusat::treeType::minId, gpusat::treeType::numSolutions, and gpusat::treeType::size.

Here is the call graph for this function:



4.1.2.11 solJson()

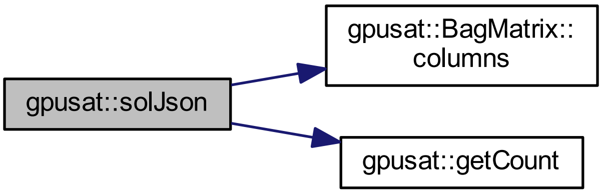
```
TableLines gpusat::solJson (
            bagType node,
            dataStructure solutionType )
```

Construct Tablelines from a solved node of the tree decomposition If no solution is stored in this node return empty grid and totalSol -1.

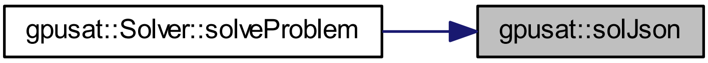
References gpusat::BagMatrix::columns(), gpusat::bagType::correction, gpusat::treeType::elements, getCount(), gpusat::TableLines::headline, gpusat::treeType::minId, gpusat::bagType::solution, gpusat::TableLines::solutions, gpusat::TableLines::totalSol, TREE, and gpusat::bagType::variables.

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:



Here is the caller graph for this function:



4.1.2.12 solutiontable()

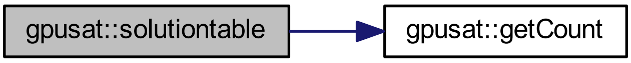
```
std::string gpusat::solutiontable (
            bagType node,
            dataStructure solutionType )  [inline]
```

Generate a formatted stringoutput for a solved node with solutions.

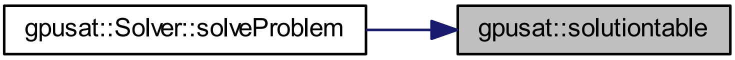
References gpusat::bagType::correction, gpusat::treeType::elements, getCount(), gpusat::treeType::minId, gpusat::bagType::solution, TREE, and gpusat::bagType::variables.

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:



Here is the caller graph for this function:



Chapter 5

Class Documentation

5.1 gpusat::BagMatrix Class Reference

```
#include <visualization.h>
```

Public Member Functions

- BagMatrix ()
- BagMatrix (size_t rows, size_t cols)
- size_t columns ()
- cl_long & operator() (size_t i, size_t j)
- cl_long operator() (size_t i, size_t j) const
- size_t rows ()

5.1.1 Detailed Description

Class for storing 2dim 'ulong' values using vector and a 1dim memory layout. See https://stackoverflow.com/a/2076668 Example usage:

BagMatrix mygrid (5, 5); for (int r = 0; r < mygrid.rows(); r++){ for (int c = 0; c < mygrid.columns(); c++){ mygrid[r][c] = r ∗ c; } } Print the array for (int r = 0; r < mygrid.rows(); r++){ std::cout << std::hex << &(mygrid[r][0]) << std::dec; std::cout << ": "; for (int c = 0; c < mygrid.columns(); c++){ std::cout << mygrid[r][c] << ' '; } std::cout << std::endl; }

5.1.2 Constructor & Destructor Documentation

5.1.2.1 BagMatrix() [1/2]

```
gpusat::BagMatrix::BagMatrix (
            size_t rows,
            size_t cols )
```

5.1.2.2 BagMatrix() [2/2]

```
gpusat::BagMatrix::BagMatrix ( )
```

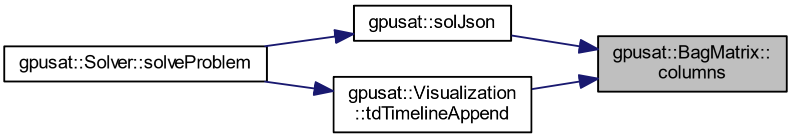
5.1.3 Member Function Documentation

5.1.3.1 columns()

```
size_t gpusat::BagMatrix::columns ( )  [inline]
```

Referenced by gpusat::solJson(), and gpusat::Visualization::tdTimelineAppend().

Here is the caller graph for this function:


gpusat::Solver::solveProblem
gpusat::solJson
gpusat::Visualization ::tdTimelineAppend
gpusat::BagMatrix:: columns

5.1.3.2 operator()() [1/2]

```
cl_long & gpusat::BagMatrix::operator() (
            size_t i,
            size_t j )
```

5.1.3.3 operator()() [2/2]

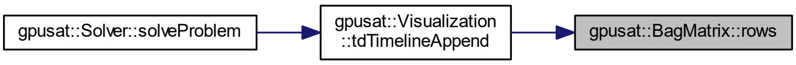
```
cl_long gpusat::BagMatrix::operator() (
            size_t i,
            size_t j ) const
```

5.1.3.4 rows()

```
size_t gpusat::BagMatrix::rows ( )  [inline]
```

Referenced by gpusat::Visualization::tdTimelineAppend().

Here is the caller graph for this function:



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

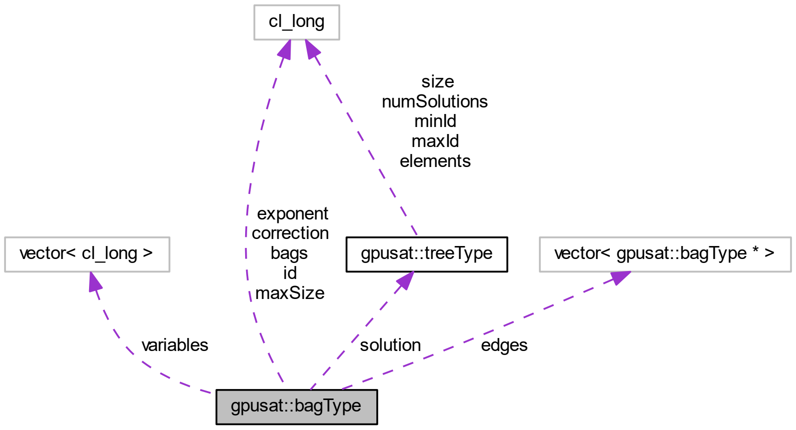
- include/visualization.h
- src/visualization.cpp

5.2 gpusat::bagType Struct Reference

type for a bag in the tree decomposition

```
#include <types.h>
```

Collaboration diagram for gpusat::bagType:



Public Attributes

- cl_long bags = 0
- cl_long correction = 0
- std::vector< bagType ∗ > edges
- cl_long exponent = 0
- cl_long id = 0
- cl_long maxSize = 0
- treeType ∗ solution
- std::vector< cl_long > variables

5.2.1 Detailed Description

type for a bag in the tree decomposition

5.2.2 Member Data Documentation

5.2.2.1 bags

```
cl_long gpusat::bagType::bags = 0
```

Referenced by gpusat::printbagType(), gpusat::Solver::solveIntroduceForget(), gpusat::Solver::solveJoin(), and gpusat::Solver::solveProblem().

5.2.2.2 correction

```
cl_long gpusat::bagType::correction = 0
```

Referenced by gpusat::printbagType(), gpusat::solJson(), gpusat::solutiontable(), gpusat::Solver::solveIntroduceForget(), and gpusat::Solver::solveJoin().

5.2.2.3 edges

```
std::vector<bagType *> gpusat::bagType::edges
```

Referenced by gpusat::Preprocessor::preprocessDecomp(), gpusat::printbagType(), gpusat::Solver::solveProblem(), and gpusat::Visualization::visuTreeDec().

5.2.2.4 exponent

```
cl_long gpusat::bagType::exponent = 0
```

Referenced by gpusat::Solver::cleanTree(), gpusat::printbagType(), gpusat::Solver::solveIntroduceForget(), and gpusat::Solver::solveJoin().

5.2.2.5 id

```
cl_long gpusat::bagType::id = 0
```

Referenced by gpusat::compTreedType(), gpusat::printbagType(), gpusat::Solver::solveIntroduceForget(), gpusat↩ ::Solver::solveProblem(), and gpusat::Visualization::visuTreeDec().

5.2.2.6 maxSize

```
cl_long gpusat::bagType::maxSize = 0
```

Referenced by gpusat::Solver::solveIntroduceForget(), gpusat::Solver::solveJoin(), and gpusat::Solver::solve↩ Problem().

5.2.2.7 solution

```
treeType* gpusat::bagType::solution
```

Referenced by gpusat::solJson(), gpusat::solutiontable(), gpusat::Solver::solveIntroduceForget(), gpusat::Solver↩ ::solveJoin(), and gpusat::Solver::solveProblem().

5.2.2.8 variables

```
std::vector<cl_long> gpusat::bagType::variables
```

Referenced by main(), gpusat::Preprocessor::preprocessDecomp(), gpusat::printbagType(), gpusat::solJson(), gpusat::solutiontable(), gpusat::Solver::solveIntroduceForget(), gpusat::Solver::solveJoin(), and gpusat::Solver↩ ::solveProblem().

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

- include/types.h

5.3 gpusat::CNFParser Class Reference

```
#include <gpusatparser.h>
```

Public Member Functions

- CNFParser (bool weighted)
- satformulaType parseSatFormula (std::string formula)

5.3.1 Constructor & Destructor Documentation

5.3.1.1 CNFParser()

```
gpusat::CNFParser::CNFParser (
            bool weighted )
```

Constructor for an CNF parser.

Parameters

weighted	indicates if weights should be assiciated with literals

5.3.2 Member Function Documentation

5.3.2.1 parseSatFormula()

```
satformulaType gpusat::CNFParser::parseSatFormula (
            std::string formula )
```

generates a sat formula from a given string

Parameters

formula	the string representation of the sat formula

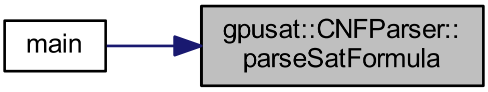
Returns

the sat formula

References gpusat::satformulaType::numVars, gpusat::satformulaType::numWeights, and gpusat::satformulaType::variableWeights.

Referenced by main().

Here is the caller graph for this function:



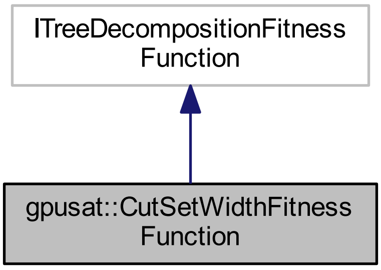
The documentation for this class was generated from the following files:

- include/gpusatparser.h
- src/gpusatparser.cpp

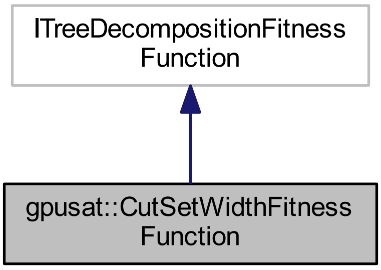
5.4 gpusat::CutSetWidthFitnessFunction Class Reference

```
#include <CutSetWidthFitnessFunction.h>
```

Inheritance diagram for gpusat::CutSetWidthFitnessFunction:



Collaboration diagram for gpusat::CutSetWidthFitnessFunction:



Public Member Functions

- CutSetWidthFitnessFunction ()=default
- ∼CutSetWidthFitnessFunction ()=default
- CutSetWidthFitnessFunction ∗ clone (void) const override
- htd::FitnessEvaluation ∗ fitness (const htd::IMultiHypergraph &graph, const htd::ITreeDecomposition &decomposition) const override
- double getMaxCutSetSize (const htd::ITreeDecomposition &decomposition) const

5.4.1 Detailed Description

Fitness function that first minimizes the cut set size and then the width.

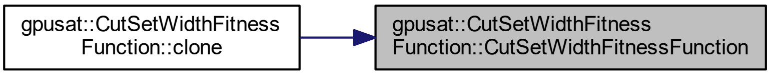
5.4.2 Constructor & Destructor Documentation

5.4.2.1 CutSetWidthFitnessFunction()

```
gpusat::CutSetWidthFitnessFunction::CutSetWidthFitnessFunction ( )  [default]
```

Referenced by clone().

Here is the caller graph for this function:


gpusat::CutSetWidthFitness
Function::clone
gpusat::CutSetWidthFitness
Function::CutSetWidthFitnessFunction

5.4.2.2 ∼CutSetWidthFitnessFunction()

```
gpusat::CutSetWidthFitnessFunction::∼CutSetWidthFitnessFunction ( )  [default]
```

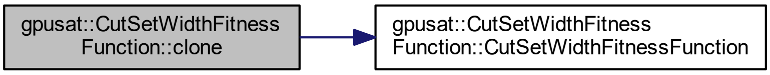
5.4.3 Member Function Documentation

5.4.3.1 clone()

```
CutSetWidthFitnessFunction ∗ gpusat::CutSetWidthFitnessFunction::clone (
             void  ) const  [override]
```

References CutSetWidthFitnessFunction().

Here is the call graph for this function:

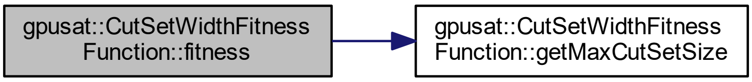

gpusat::CutSetWidthFitness
Function::clone
gpusat::CutSetWidthFitness
Function::CutSetWidthFitnessFunction

5.4.3.2 fitness()

```
htd::FitnessEvaluation ∗ gpusat::CutSetWidthFitnessFunction::fitness (
             const htd::IMultiHypergraph & graph,
             const htd::ITreeDecomposition & decomposition ) const  [override]
```

References getMaxCutSetSize().

Here is the call graph for this function:

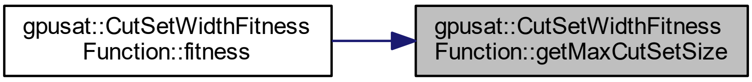

gpusat::CutSetWidthFitness
Function::fitness
gpusat::CutSetWidthFitness
Function::getMaxCutSetSize

5.4.3.3 getMaxCutSetSize()

```
double gpusat::CutSetWidthFitnessFunction::getMaxCutSetSize (
            const htd::ITreeDecomposition & decomposition ) const
```

Referenced by fitness().

Here is the caller graph for this function:


gpusat::CutSetWidthFitness Function::fitness
gpusat::CutSetWidthFitness Function::getMaxCutSetSize

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

- include/FitnessFunctions/CutSetWidthFitnessFunction.h
- src/FitnessFunctions/CutSetWidthFitnessFunction.cpp

5.5 gpusat::Decomposer Class Reference

```
#include <decomposer.h>
```

Static Public Member Functions

- static std::string computeDecomposition (std::string formula, htd::ITreeDecompositionFitnessFunction *fitness, size_t n)

5.5.1 Member Function Documentation

5.5.1.1 computeDecomposition()

```
std::string gpusat::Decomposer::computeDecomposition (
            std::string formula,
            htd::ITreeDecompositionFitnessFunction * fitness,
            size_t n ) [static]
```

computes the decomposition of the primal graph of the given formula

Parameters

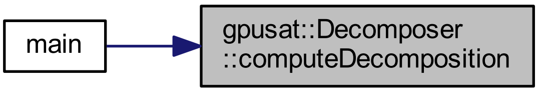
formula	the formula in cnf format
fitness	the fitness function
n	number of iterations for the fitness function

Returns

the decomposition in td format

Referenced by main().

Here is the caller graph for this function:



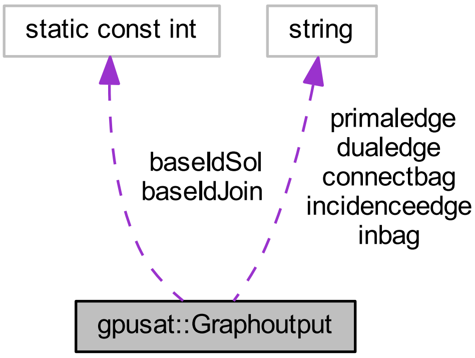
The documentation for this class was generated from the following files:

- include/decomposer.h
- src/decomposer.cpp

5.6 gpusat::Graphoutput Class Reference

```
#include <graphoutput.h>
```

Collaboration diagram for gpusat::Graphoutput:



Public Member Functions

- Graphoutput (std::string filename)
- std::string getFilename ()
- void graphEdgeSet (treedecType ∗dec)

    Addes the edges from the tree-decomposition into the graph.
- void graphEnd ()
- void graphStart (treedecType ∗dec)
- bool isEnabled ()
- void neo4jSat (satformulaType ∗satFormula)

    Output Cypherquery to create the nodes from the SAT formula
- void neo4jTD (treedecType ∗treeDec)
- void nodeBag (unsigned int id, std::string solution)

    Create one node with the corresponding solution in a connected extra bag.
- void nodeJoin (unsigned int id1, unsigned int id2, std::string solution)

    Creates a node with the solution that joins two bags. Two edges are added to connect the solution.

Static Public Attributes

- static const int baseIdJoin = 4 ∗ baseIdSol
- static const int baseIdSol = 1'000'000
- static const std::string connectbag = "USES_BAG"
- static const std::string dualedge = "SHARE_VAR"
- static const std::string inbag = "CONTAINS"
- static const std::string incidenceedge = "VAR_IN_CLAUSE"
- static const std::string primaledge = "SHARE_CLAUSE"

5.6.1 Constructor & Destructor Documentation

5.6.1.1 Graphoutput()

```
gpusat::Graphoutput::Graphoutput (
            std::string filename )  [inline]
```

5.6.2 Member Function Documentation

5.6.2.1 getFilename()

```
std::string gpusat::Graphoutput::getFilename ( )  [inline]
```

5.6.2.2 graphEdgeSet()

```
void gpusat::Graphoutput::graphEdgeSet (
            treedecType ∗ dec )
```

Addes the edges from the tree-decomposition into the graph.

Any outgoing edges from id '0' are not added.

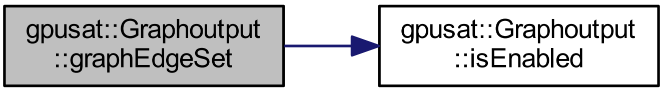
Parameters

dec	Pointer to the tree-decomposition containing used edges.

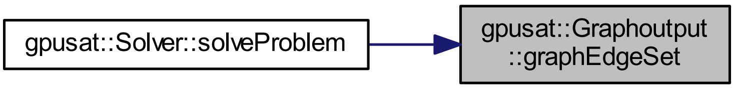
References gpusat::treedecType::bags, and isEnabled().

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:


gpusat::Graphoutput ::graphEdgeSet
gpusat::Graphoutput ::isEnabled

Here is the caller graph for this function:


gpusat::Solver::solveProblem
gpusat::Graphoutput ::graphEdgeSet

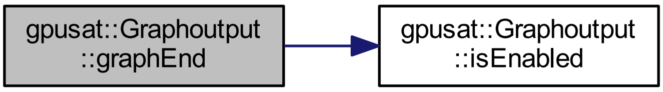
5.6.2.3 graphEnd()

```
void gpusat::Graphoutput::graphEnd ( )
```

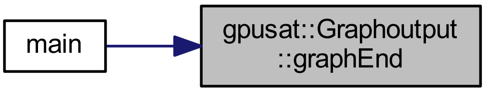
References isEnabled().

Referenced by main().

Here is the call graph for this function:


gpusat::Graphoutput ::graphEnd
gpusat::Graphoutput ::isEnabled

Here is the caller graph for this function:



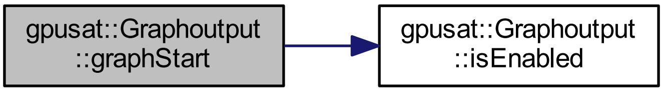
5.6.2.4 graphStart()

```
void gpusat::Graphoutput::graphStart (
            treedecType * dec )
```

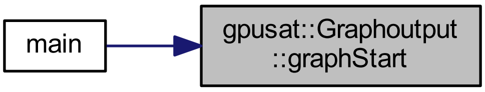
References gpusat::treedecType::bags, and isEnabled().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:

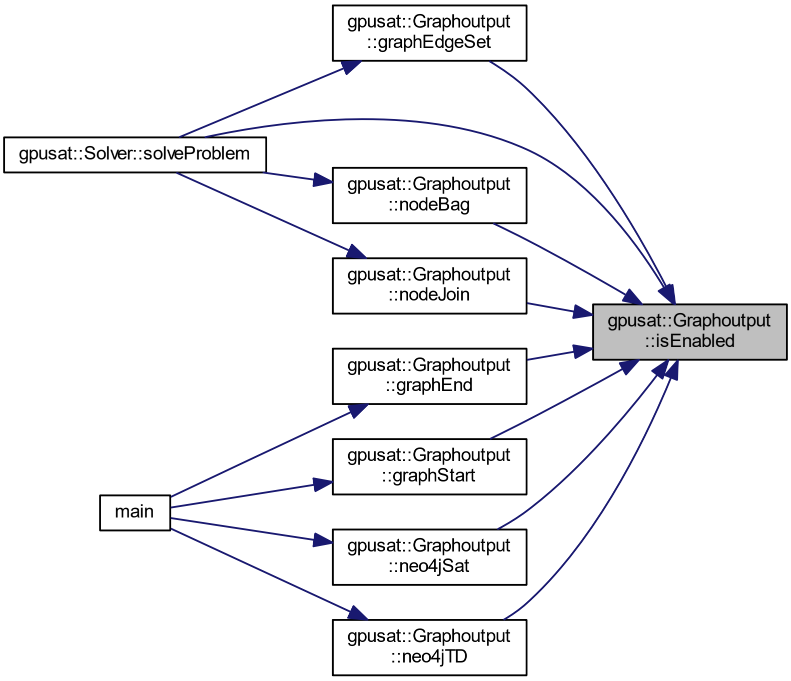


5.6.2.5 isEnabled()

```
bool gpusat::Graphoutput::isEnabled ( )  [inline]
```

Referenced by graphEdgeSet(), graphEnd(), graphStart(), neo4jSat(), neo4jTD(), nodeBag(), nodeJoin(), and gpusat::Solver::solveProblem().

Here is the caller graph for this function:


gpusat::Graphoutput ::graphEdgeSet
gpusat::Solver::solveProblem
gpusat::Graphoutput ::nodeBag
gpusat::Graphoutput ::nodeJoin
gpusat::Graphoutput ::isEnabled
gpusat::Graphoutput ::graphEnd
gpusat::Graphoutput ::graphStart
main
gpusat::Graphoutput ::neo4jSat
gpusat::Graphoutput ::neo4jTD

5.6.2.6 neo4jSat()

```
void gpusat::Graphoutput::neo4jSat (
            satformulaType * satFormula )
```

Output Cypherquery to create the nodes from the SAT formula

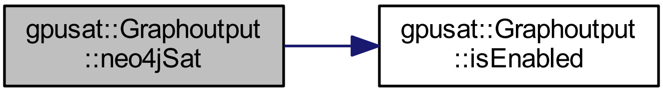
Parameters

satFormula	The sat formula.

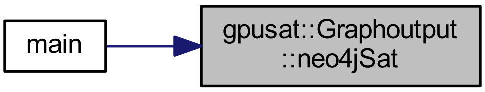
References gpusat::satformulaType::clauses, dualedge, gpusat::satformulaType::facts, incidenceedge, isEnabled(), gpusat::satformulaType::numVars, primaledge, and gpusat::satformulaType::unsat.

Referenced by main().

Here is the call graph for this function:


gpusat::Graphoutput ::neo4jSat
gpusat::Graphoutput ::isEnabled

Here is the caller graph for this function:



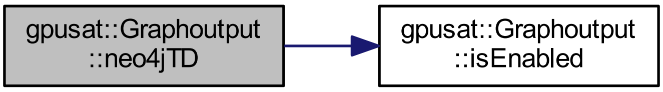
5.6.2.7 neo4jTD()

```
void gpusat::Graphoutput::neo4jTD (
            treedecType * treeDec )
```

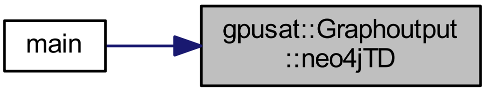
References gpusat::treedecType::bags, connectbag, inbag, isEnabled(), gpusat::treedecType::numb, gpusat::treedecType::numVars, and gpusat::treedecType::width.

Referenced by main().

Here is the call graph for this function:


gpusat::Graphoutput ::neo4jTD
gpusat::Graphoutput ::isEnabled

Here is the caller graph for this function:



5.6.2.8 nodeBag()

```
void gpusat::Graphoutput::nodeBag (
            unsigned int id,
            std::string solution )
```

Create one node with the corresponding solution in a connected extra bag.

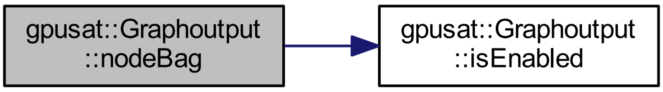
Parameters

id	The identifier of the bag.
solution	The solution in string form.

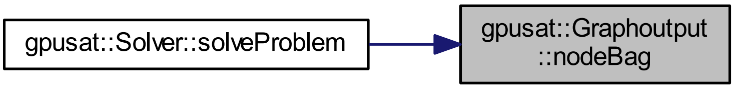
References isEnabled().

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:



Here is the caller graph for this function:



5.6.2.9 nodeJoin()

```
void gpusat::Graphoutput::nodeJoin (
             unsigned int id1,
             unsigned int id2,
             std::string solution )
```

Creates a node with the solution that joins two bags. Two edges are added to connect the solution.

Parameters

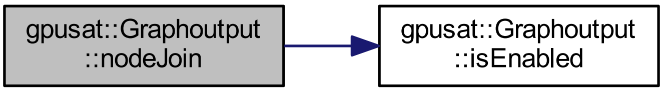
id1	The id1.
id2	The id2.
solution	The solution.

TODO Edit XML Comment Template for nodeJoin

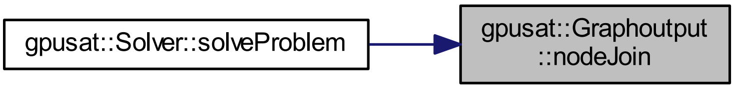
References isEnabled().

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:


gpusat::Graphoutput ::nodeJoin
gpusat::Graphoutput ::isEnabled

Here is the caller graph for this function:



5.6.3 Member Data Documentation

5.6.3.1 baseIdJoin

```
const int gpusat::Graphoutput::baseIdJoin = 4 * baseIdSol  [static]
```

5.6.3.2 baseIdSol

```
const int gpusat::Graphoutput::baseIdSol = 1'000'000  [static]
```

5.6.3.3 connectbag

```
const std::string gpusat::Graphoutput::connectbag = "USES_BAG"  [static]
```

Referenced by neo4jTD().

5.6.3.4 dualedge

```
const std::string gpusat::Graphoutput::dualedge = "SHARE_VAR"  [static]
```

Referenced by neo4jSat().

5.6.3.5 inbag

```
const std::string gpusat::Graphoutput::inbag = "CONTAINS"  [static]
```

Referenced by neo4jTD().

5.6.3.6 incidenceedge

```
const std::string gpusat::Graphoutput::incidenceedge = "VAR_IN_CLAUSE"  [static]
```

Referenced by neo4jSat().

5.6.3.7 primaledge

```
const std::string gpusat::Graphoutput::primaledge = "SHARE_CLAUSE"  [static]
```

Referenced by neo4jSat().

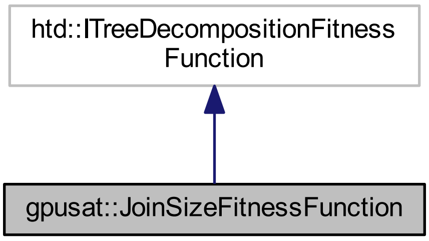
The documentation for this class was generated from the following files:

- include/graphoutput.h
- src/graphoutput.cpp

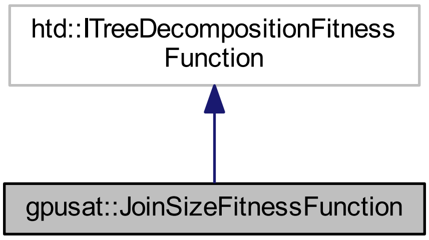
5.7 gpusat::JoinSizeFitnessFunction Class Reference

```
#include <JoinSizeFitnessFunction.h>
```

Inheritance diagram for gpusat::JoinSizeFitnessFunction:


htd::ITreeDecompositionFitness Function
gpusat::JoinSizeFitnessFunction

Collaboration diagram for gpusat::JoinSizeFitnessFunction:


htd::ITreeDecompositionFitness Function
gpusat::JoinSizeFitnessFunction

Public Member Functions

- JoinSizeFitnessFunction ()=default
- ∼JoinSizeFitnessFunction ()=default
- JoinSizeFitnessFunction ∗ clone (void) const override
- htd::FitnessEvaluation ∗ fitness (const htd::IMultiHypergraph &graph, const htd::ITreeDecomposition &decomposition) const override

5.7.1 Detailed Description

Fitness function that minimizes the number of vertices in a join node and then the width.

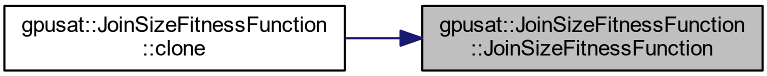
5.7.2 Constructor & Destructor Documentation

5.7.2.1 JoinSizeFitnessFunction()

```
gpusat::JoinSizeFitnessFunction::JoinSizeFitnessFunction ( )  [default]
```

Referenced by clone().

Here is the caller graph for this function:


gpusat::JoinSizeFitnessFunction ::clone
gpusat::JoinSizeFitnessFunction ::JoinSizeFitnessFunction

5.7.2.2 ∼JoinSizeFitnessFunction()

```
gpusat::JoinSizeFitnessFunction::∼JoinSizeFitnessFunction ( )  [default]
```

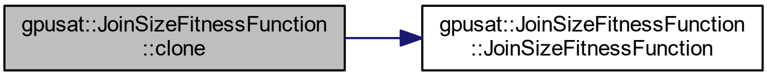
5.7.3 Member Function Documentation

5.7.3.1 clone()

```
JoinSizeFitnessFunction * gpusat::JoinSizeFitnessFunction::clone (
             void  ) const  [override]
```

References JoinSizeFitnessFunction().

Here is the call graph for this function:


gpusat::JoinSizeFitnessFunction ::clone
gpusat::JoinSizeFitnessFunction ::JoinSizeFitnessFunction

5.7.3.2 fitness()

```
htd::FitnessEvaluation * gpusat::JoinSizeFitnessFunction::fitness (
             const htd::IMultiHypergraph & graph,
             const htd::ITreeDecomposition & decomposition ) const  [override]
```

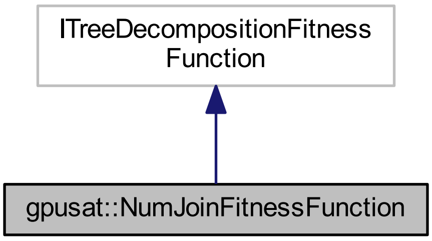
The documentation for this class was generated from the following files:

- include/FitnessFunctions/JoinSizeFitnessFunction.h
- src/FitnessFunctions/JoinSizeFitnessFunction.cpp

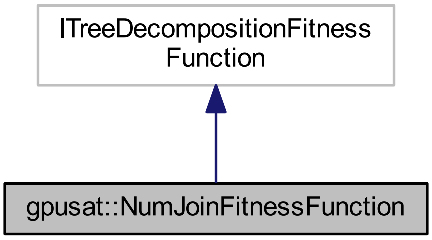
5.8 gpusat::NumJoinFitnessFunction Class Reference

```
#include <NumJoinFitnessFunction.h>
```

Inheritance diagram for gpusat::NumJoinFitnessFunction:


ITreeDecompositionFitness Function
gpusat::NumJoinFitnessFunction

Collaboration diagram for gpusat::NumJoinFitnessFunction:


ITreeDecompositionFitness Function
gpusat::NumJoinFitnessFunction

Public Member Functions

- NumJoinFitnessFunction ()=default
- ∼NumJoinFitnessFunction ()=default
- NumJoinFitnessFunction ∗ clone (void) const override
- htd::FitnessEvaluation ∗ fitness (const htd::IMultiHypergraph &graph, const htd::ITreeDecomposition &decomposition) const override

5.8.1 Detailed Description

Fitness function that minimizes the number of join nodes and then the width.

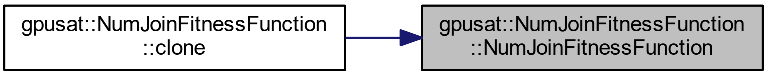
5.8.2 Constructor & Destructor Documentation

5.8.2.1 NumJoinFitnessFunction()

```
gpusat::NumJoinFitnessFunction::NumJoinFitnessFunction ( )  [default]
```

Referenced by clone().

Here is the caller graph for this function:


gpusat::NumJoinFitnessFunction ::clone
gpusat::NumJoinFitnessFunction ::NumJoinFitnessFunction

5.8.2.2 ∼NumJoinFitnessFunction()

```
gpusat::NumJoinFitnessFunction::∼NumJoinFitnessFunction ( )  [default]
```

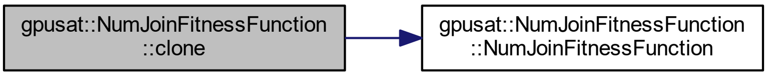
5.8.3 Member Function Documentation

5.8.3.1 clone()

```
NumJoinFitnessFunction * gpusat::NumJoinFitnessFunction::clone (
            void  ) const  [override]
```

References NumJoinFitnessFunction().

Here is the call graph for this function:


gpusat::NumJoinFitnessFunction ::clone
gpusat::NumJoinFitnessFunction ::NumJoinFitnessFunction

5.8.3.2 fitness()

```
htd::FitnessEvaluation * gpusat::NumJoinFitnessFunction::fitness (
            const htd::IMultiHypergraph & graph,
            const htd::ITreeDecomposition & decomposition ) const  [override]
```

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

- include/FitnessFunctions/NumJoinFitnessFunction.h
- src/FitnessFunctions/NumJoinFitnessFunction.cpp

5.9 gpusat::Preprocessor Class Reference

```
#include <gpusatpreprocessor.h>
```

Static Public Member Functions

- static void preprocessDecomp (bagType *decomp, cl_long combineWidth)
- static void preprocessFacts (treedecType &decomp, satformulaType &formula, cl_double &defaultWeight)

5.9.1 Member Function Documentation

5.9.1.1 preprocessDecomp()

```
void gpusat::Preprocessor::preprocessDecomp (
            bagType * decomp,
            cl_long combineWidth )  [static]
```

preprocess the tree decomposition

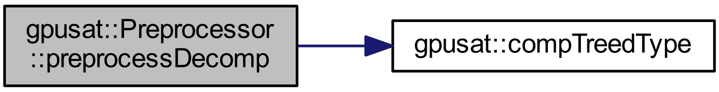
Parameters

decomp	the tree decomposition
combineWidth	max width to combine bags

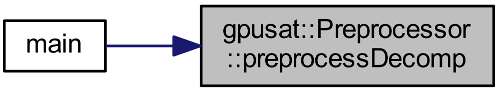
References gpusat::compTreedType(), gpusat::bagType::edges, and gpusat::bagType::variables.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.1.2 preprocessFacts()

```
void gpusat::Preprocessor::preprocessFacts (
            treedecType & decomp,
            satformulaType & formula,
            cl_double & defaultWeight )  [static]
```

removes facts from the sat formula

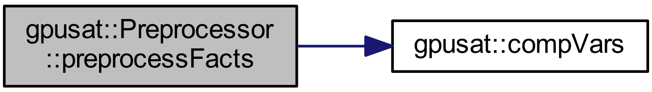
Parameters

decomp	the tree decomposition
formula	the sat formula
defaultWeight	for WMC the product of the weights of the removed literals

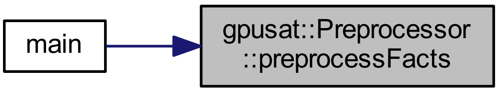
References gpusat::treedecType::bags, gpusat::satformulaType::clauses, gpusat::compVars(), gpusat::satformulaType::facts, gpusat::treedecType::numVars, gpusat::satformulaType::numVars, gpusat::satformulaType::numWeights, gpusat::satformulaType::unsat, and gpusat::satformulaType::variableWeights.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



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

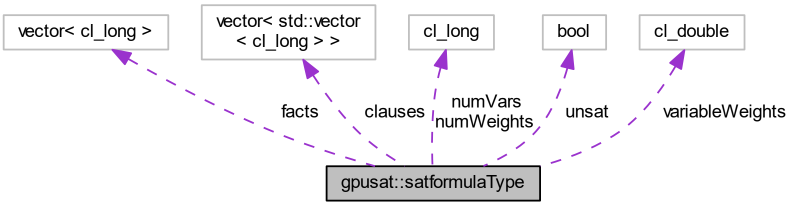
- include/gpusatpreprocessor.h
- src/gpusatpreprocessor.cpp

5.10 gpusat::satformulaType Struct Reference

type for saving the sat formula

```
#include <types.h>
```

Collaboration diagram for gpusat::satformulaType:



Public Attributes

- std::vector< std::vector< cl_long > > clauses
- std::vector< cl_long > facts
- cl_long numVars = 0
- cl_long numWeights = 0
- bool unsat = false
- cl_double * variableWeights = nullptr

5.10.1 Detailed Description

type for saving the sat formula

5.10.2 Member Data Documentation

5.10.2.1 clauses

```
std::vector<std::vector<cl_long> > gpusat::satformulaType::clauses
```

Referenced by gpusat::Graphoutput::neo4jSat(), gpusat::Preprocessor::preprocessFacts(), gpusat::Solver::solve↩
IntroduceForget(), and gpusat::Visualization::visuClauses().

5.10.2.2 facts

```
std::vector<cl_long> gpusat::satformulaType::facts
```

Referenced by gpusat::Graphoutput::neo4jSat(), and gpusat::Preprocessor::preprocessFacts().

5.10.2.3 numVars

```
cl_long gpusat::satformulaType::numVars = 0
```

Referenced by gpusat::Graphoutput::neo4jSat(), gpusat::CNFParser::parseSatFormula(), and gpusat::↩
Preprocessor::preprocessFacts().

5.10.2.4 numWeights

```
cl_long gpusat::satformulaType::numWeights = 0
```

Referenced by gpusat::CNFParser::parseSatFormula(), gpusat::Preprocessor::preprocessFacts(), gpusat::Solver↩ ::solveIntroduceForget(), and gpusat::Solver::solveJoin().

5.10.2.5 unsat

```
bool gpusat::satformulaType::unsat = false
```

Referenced by main(), gpusat::Graphoutput::neo4jSat(), and gpusat::Preprocessor::preprocessFacts().

5.10.2.6 variableWeights

```
cl_double* gpusat::satformulaType::variableWeights = nullptr
```

Referenced by gpusat::CNFParser::parseSatFormula(), gpusat::Preprocessor::preprocessFacts(), gpusat::Solver↩ ::solveIntroduceForget(), and gpusat::Solver::solveJoin().

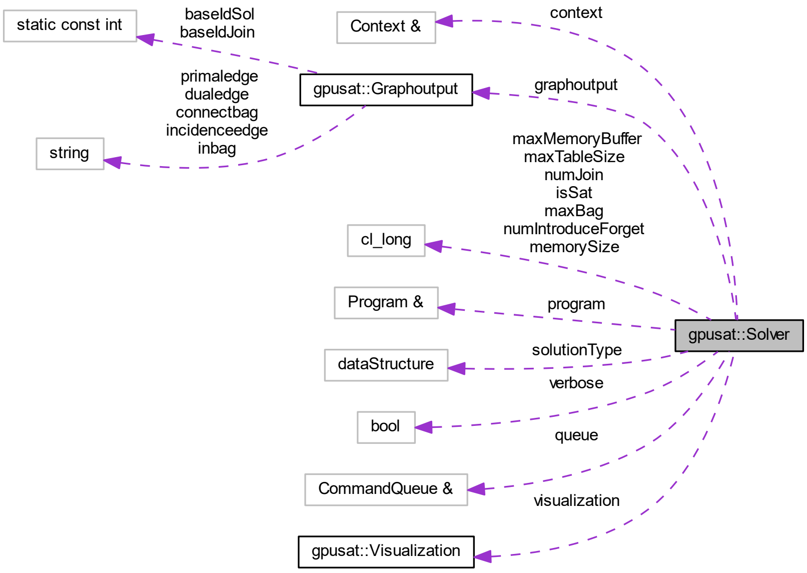
The documentation for this struct was generated from the following file:

- include/types.h

5.11 gpusat::Solver Class Reference

```
#include <solver.h>
```

Collaboration diagram for gpusat::Solver:



Public Member Functions

- Solver (cl::Context &context_, cl::CommandQueue &queue_, cl::Program &program_, cl_long memorySize↵ _, cl_long maxMemoryBuffer_, dataStructure solutionType_, cl_long maxBag_, bool verbose_, Graphoutput ∗graphoutput_, Visualization ∗visualization_)
- void solveProblem (treedecType &decomp, satformulaType &formula, bagType &node, bagType &pnode, nodeType lastNode)

  Solves the sat formula with a given decomposition.

Public Attributes

- Graphoutput ∗ graphoutput
- cl_long isSat = 1
- cl_long maxBag = 0
- cl_long maxMemoryBuffer = 0
- cl_long maxTableSize = 0
- cl_long numIntroduceForget = 0
- cl_long numJoin = 0
- dataStructure solutionType = dataStructure::TREE
- bool verbose = false
- Visualization ∗ visualization

Protected Member Functions

- void cleanTree (treeType &table, cl_long size, cl_long numVars, bagType &node, cl_long nextSize)
- void combineTree (treeType &to, treeType &from, cl_long numVars)
- void solveIntroduceForget (satformulaType &formula, bagType &pnode, bagType &node, bagType &cnode, bool leaf, nodeType nextNode)
- void solveJoin (bagType &node, bagType &edge1, bagType &edge2, satformulaType &formula, nodeType nextNode)

Protected Attributes

- cl::Context & context
- cl_long memorySize
- cl::Program & program
- cl::CommandQueue & queue

5.11.1 Detailed Description

Organizes the process of solving a sat problem in the given context.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 Solver()

```
gpusat::Solver::Solver (
            cl::Context & context_,
            cl::CommandQueue & queue_,
            cl::Program & program_,
            cl_long memorySize_,
            cl_long maxMemoryBuffer_,
            dataStructure solutionType_,
            cl_long maxBag_,
            bool verbose_,
            Graphoutput * graphoutput_,
            Visualization * visualization_ )  [inline]
```

Parameters

*context_*	the given context as hardware
*queue_*	
*program_*	
*memorySize_*	
*maxMemory↵ Buffer_*	
*solutionType_*	
*maxBag_*	
*verbose_*	
*graphoutput_*	
*visualization_*	

5.11.3 Member Function Documentation

5.11.3.1 cleanTree()

```
void gpusat::Solver::cleanTree (
            treeType & table,
            cl_long size,
            cl_long numVars,
            bagType & node,
            cl_long nextSize )  [protected]
```

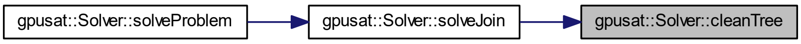
Parameters

table	
size	
numVars	
node	

References context, gpusat::treeType::elements, gpusat::bagType::exponent, gpusat::treeType::maxId, gpusat↵ ::treeType::minId, gpusat::treeType::numSolutions, program, queue, and gpusat::treeType::size.

Referenced by solveJoin().

Here is the caller graph for this function:


gpusat::Solver::solveProblem
gpusat::Solver::solveJoin
gpusat::Solver::cleanTree

5.11.3.2 combineTree()

```
void gpusat::Solver::combineTree (
             treeType & to,
             treeType & from,
             cl_long numVars )  [protected]
```

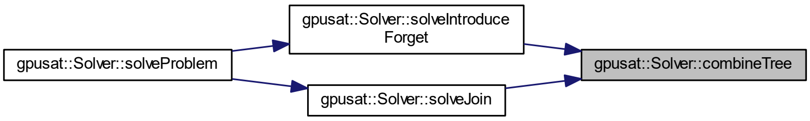
Parameters

to	
from	
numVars	

References context, gpusat::treeType::elements, gpusat::treeType::maxId, gpusat::treeType::minId, gpusat::treeType::numSolutions, program, queue, and gpusat::treeType::size.

Referenced by solveIntroduceForget(), and solveJoin().

Here is the caller graph for this function:


gpusat::Solver::solveProblem
gpusat::Solver::solveIntroduce Forget
gpusat::Solver::solveJoin
gpusat::Solver::combineTree

5.11.3.3 solveIntroduceForget()

```
void gpusat::Solver::solveIntroduceForget (
             satformulaType & formula,
             bagType & pnode,
             bagType & node,
             bagType & cnode,
             bool leaf,
             nodeType nextNode )  [protected]
```

function to solve an introduce forget node

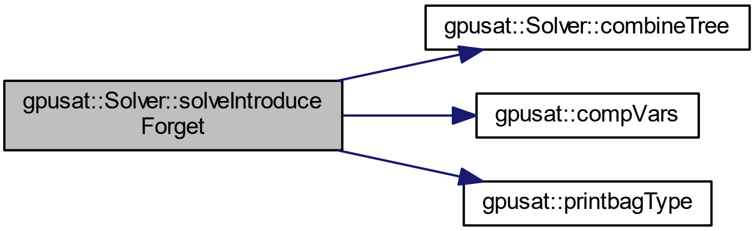
Parameters

formula	the sat formula
pnode	the parent of the current node
node	the current node
cnode	the child of the current node
leaf	indicates that the current node is a leaf node

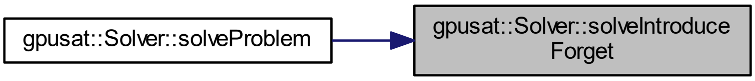
References gpusat::ARRAY, gpusat::bagType::bags, gpusat::satformulaType::clauses, combineTree(), gpusat::compVars(), context, gpusat::bagType::correction, gpusat::treeType::elements, gpusat::bagType::exponent, gpusat::bagType::id, gpusat::INTRODUCEFORGET, isSat, gpusat::JOIN, maxBag, gpusat::treeType::maxId, maxMemoryBuffer, gpusat::bagType::maxSize, maxTableSize, memorySize, gpusat::treeType::minId, numIntroduceForget, gpusat::treeType::numSolutions, gpusat::satformulaType::numWeights, gpusat::printbagType(), program, queue, gpusat::treeType::size, gpusat::bagType::solution, solutionType, gpusat::TREE, gpusat::bagType::variables, gpusat::satformulaType::variableWeights, and verbose.

Referenced by solveProblem().

Here is the call graph for this function:


gpusat::Solver::combineTree
gpusat::Solver::solveIntroduce Forget
gpusat::compVars
gpusat::printbagType

Here is the caller graph for this function:


gpusat::Solver::solveProblem
gpusat::Solver::solveIntroduce Forget

5.11.3.4 solveJoin()

```
void gpusat::Solver::solveJoin (
             bagType & node,
             bagType & edge1,
             bagType & edge2,
             satformulaType & formula,
             nodeType nextNode )  [protected]
```

function to solve a join node

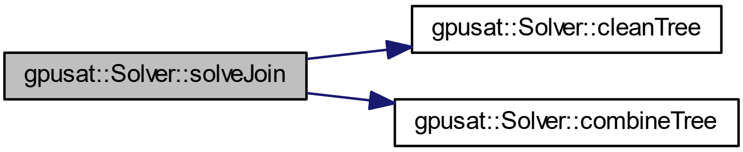
Parameters

node	the node to save the solutions in
edge1	the first child node
edge2	the second child node
formula	the sat formula

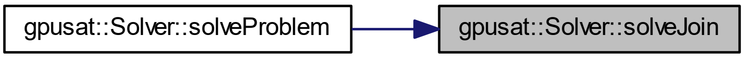
References gpusat::ARRAY, gpusat::bagType::bags, cleanTree(), combineTree(), context, gpusat::bagType::correction, gpusat::treeType::elements, gpusat::bagType::exponent, gpusat::INTRODUCEFORGET, isSat, gpusat::JOIN, maxBag, gpusat::treeType::maxId, maxMemoryBuffer, gpusat::bagType::maxSize, maxTableSize, memorySize, gpusat::treeType::minId, numJoin, gpusat::treeType::numSolutions, gpusat::satformulaType::numWeights, program, queue, gpusat::treeType::size, gpusat::bagType::solution, solutionType, gpusat::TREE, gpusat::bagType::variables, and gpusat::satformulaType::variableWeights.

Referenced by solveProblem().

Here is the call graph for this function:


gpusat::Solver::solveJoin
gpusat::Solver::cleanTree
gpusat::Solver::combineTree

Here is the caller graph for this function:



5.11.3.5 solveProblem()

```
void gpusat::Solver::solveProblem (
            treedecType & decomp,
            satformulaType & formula,
            bagType & node,
            bagType & pnode,
            nodeType lastNode )
```

Solves the sat formula with a given decomposition.

function to solve the sat problem

Parameters

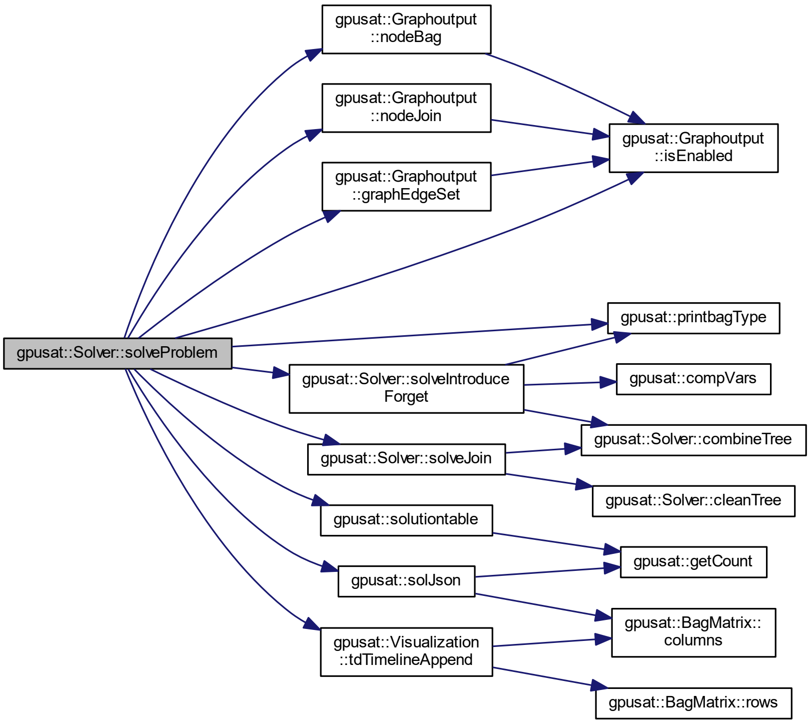
decomp	the tree decomposition
formula	the sat formula
node	the node to start from in the tree decompostion

Parameters

decomp	The decomposition of the
formula	The formula.
node	The node.
pnode	The pnode.
lastNode	The last node.

References gpusat::bagType::bags, gpusat::bagType::edges, gpusat::treeType::elements, gpusat::Graphoutput↩ ::graphEdgeSet(), graphoutput, gpusat::bagType::id, gpusat::INTRODUCEFORGET, gpusat::Graphoutput::is↩ Enabled(), isSat, gpusat::JOIN, gpusat::treeType::maxId, gpusat::bagType::maxSize, gpusat::treeType::minId, gpusat::Graphoutput::nodeBag(), gpusat::Graphoutput::nodeJoin(), gpusat::treeType::numSolutions, gpusat↩ ::printbagType(), gpusat::treeType::size, gpusat::solJson(), gpusat::bagType::solution, gpusat::solutiontable(), solutionType, solveIntroduceForget(), solveJoin(), gpusat::Visualization::tdTimelineAppend(), gpusat::bagType↩ ::variables, verbose, and visualization.

Here is the call graph for this function:


gpusat::Graphoutput ::nodeBag
gpusat::Graphoutput ::nodeJoin
gpusat::Graphoutput ::graphEdgeSet
gpusat::Graphoutput ::isEnabled
gpusat::printbagType
gpusat::Solver::solveProblem
gpusat::Solver::solveIntroduce Forget
gpusat::compVars
gpusat::Solver::combineTree
gpusat::Solver::solveJoin
gpusat::Solver::cleanTree
gpusat::solutiontable
gpusat::getCount
gpusat::solJson
gpusat::BagMatrix:: columns
gpusat::Visualization ::tdTimelineAppend
gpusat::BagMatrix::rows

5.11.4 Member Data Documentation

5.11.4.1 context

```
cl::Context& gpusat::Solver::context  [protected]
```

Referenced by cleanTree(), combineTree(), solveIntroduceForget(), and solveJoin().

5.11.4.2 graphoutput

```
Graphoutput* gpusat::Solver::graphoutput
```

Referenced by solveProblem().

5.11.4.3 isSat

```
cl_long gpusat::Solver::isSat = 1
```

Referenced by solveIntroduceForget(), solveJoin(), and solveProblem().

5.11.4.4 maxBag

```
cl_long gpusat::Solver::maxBag = 0
```

Referenced by solveIntroduceForget(), and solveJoin().

5.11.4.5 maxMemoryBuffer

```
cl_long gpusat::Solver::maxMemoryBuffer = 0
```

Referenced by solveIntroduceForget(), and solveJoin().

5.11.4.6 maxTableSize

```
cl_long gpusat::Solver::maxTableSize = 0
```

Referenced by main(), solveIntroduceForget(), and solveJoin().

5.11.4.7 memorySize

```
cl_long gpusat::Solver::memorySize  [protected]
```

Referenced by solveIntroduceForget(), and solveJoin().

5.11.4.8 numIntroduceForget

```
cl_long gpusat::Solver::numIntroduceForget = 0
```

Referenced by main(), and solveIntroduceForget().

5.11.4.9 numJoin

```
cl_long gpusat::Solver::numJoin = 0
```

Referenced by main(), and solveJoin().

5.11.4.10 program

```
cl::Program& gpusat::Solver::program  [protected]
```

Referenced by cleanTree(), combineTree(), solveIntroduceForget(), and solveJoin().

5.11.4.11 queue

```
cl::CommandQueue& gpusat::Solver::queue  [protected]
```

Referenced by cleanTree(), combineTree(), solveIntroduceForget(), and solveJoin().

5.11.4.12 solutionType

```
dataStructure gpusat::Solver::solutionType = dataStructure::TREE
```

Referenced by solveIntroduceForget(), solveJoin(), and solveProblem().

5.11.4.13 verbose

```
bool gpusat::Solver::verbose = false
```

Referenced by solveIntroduceForget(), and solveProblem().

5.11.4.14 visualization

```
Visualization* gpusat::Solver::visualization
```

Referenced by solveProblem().

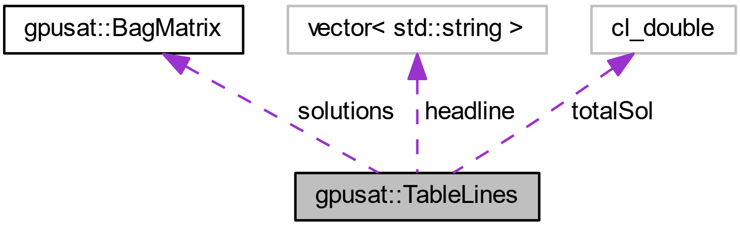
The documentation for this class was generated from the following files:

- include/solver.h
- src/solver.cpp

5.12 gpusat::TableLines Struct Reference

```
#include <visualization.h>
```

Collaboration diagram for gpusat::TableLines:



Public Attributes

- std::vector< std::string > headline
- BagMatrix solutions
- cl_double totalSol

5.12.1 Detailed Description

Solutions in Timeline. Example: [["id", "v2", "v4", "n Sol"], [0, 0, 0, 1], [1, 1, 0, 2], [2, 0, 1, 2], [3, 1, 1, 3]]

5.12.2 Member Data Documentation

5.12.2.1 headline

```
std::vector<std::string> gpusat::TableLines::headline
```

Referenced by gpusat::solJson(), and gpusat::Visualization::tdTimelineAppend().

5.12.2.2 solutions

```
BagMatrix gpusat::TableLines::solutions
```

Referenced by gpusat::solJson(), and gpusat::Visualization::tdTimelineAppend().

5.12.2.3 totalSol

```
cl_double gpusat::TableLines::totalSol
```

Referenced by gpusat::solJson(), and gpusat::Visualization::tdTimelineAppend().

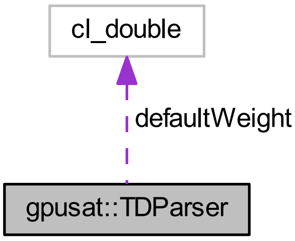
The documentation for this struct was generated from the following file:

- include/visualization.h

5.13 gpusat::TDParser Class Reference

```
#include <gpusatparser.h>
```

Collaboration diagram for gpusat::TDParser:



Public Member Functions

- TDParser ()
- treedecType parseTreeDecomp (std::string graph, satformulaType &formula)

Public Attributes

- cl_double defaultWeight = 1.0

5.13.1 Constructor & Destructor Documentation

5.13.1.1 TDParser()

```
gpusat::TDParser::TDParser ( )  [inline]
```

5.13.2 Member Function Documentation

5.13.2.1 parseTreeDecomp()

```
treedecType gpusat::TDParser::parseTreeDecomp (
             std::string graph,
             satformulaType & formula )
```

generates a treedec from a given string

Parameters

graph	the string representation of the tree decomposition

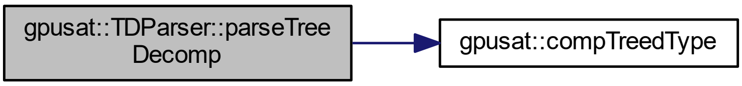
Returns

the tree decomposition

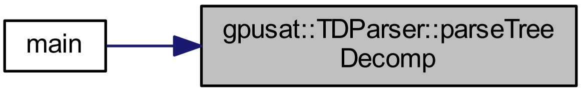
References gpusat::treedecType::bags, gpusat::compTreedType(), and gpusat::treedecType::numb.

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



5.13.3 Member Data Documentation

5.13.3.1 defaultWeight

```
cl_double gpusat::TDParser::defaultWeight = 1.0
```

Referenced by main().

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

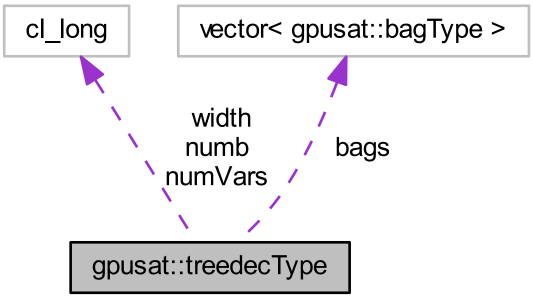
- include/gpusatparser.h
- src/gpusatparser.cpp

5.14 gpusat::treedecType Struct Reference

type for saving a tree decomposition

```
#include <types.h>
```

Collaboration diagram for gpusat::treedecType:



Public Attributes

- std::vector< bagType > bags
- cl_long numb = 0
- cl_long numVars = 0
- cl_long width = 0

5.14.1 Detailed Description

type for saving a tree decomposition

5.14.2 Member Data Documentation

5.14.2.1 bags

```
std::vector<bagType> gpusat::treedecType::bags
```

Referenced by gpusat::Graphoutput::graphEdgeSet(), gpusat::Graphoutput::graphStart(), main(), gpusat::Graphoutput::neo4jTD(), gpusat::TDParser::parseTreeDecomp(), gpusat::Preprocessor::preprocessFacts(), gpusat::printtreedecType(), and gpusat::Visualization::visuTreeDec().

5.14.2.2 numb

```
cl_long gpusat::treedecType::numb = 0
```

Referenced by gpusat::Graphoutput::neo4jTD(), gpusat::TDParser::parseTreeDecomp(), and gpusat::printtreedecType().

5.14.2.3 numVars

```
cl_long gpusat::treedecType::numVars = 0
```

Referenced by gpusat::Graphoutput::neo4jTD(), gpusat::Preprocessor::preprocessFacts(), and gpusat::↩ Visualization::visuTreeDec().

5.14.2.4 width

```
cl_long gpusat::treedecType::width = 0
```

Referenced by main(), and gpusat::Graphoutput::neo4jTD().

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

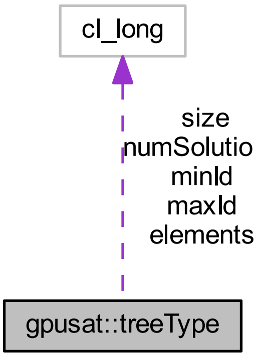
- include/types.h

5.15 gpusat::treeType Struct Reference

tree type for storing the models

```
#include <types.h>
```

Collaboration diagram for gpusat::treeType:



Public Attributes

- cl_long ∗ elements = nullptr
- cl_long maxId = 0
- cl_long minId = 0
- cl_long numSolutions = 0
- cl_long size = 0

5.15.1 Detailed Description

tree type for storing the models

5.15.2 Member Data Documentation

5.15.2.1 elements

```
cl_long* gpusat::treeType::elements = nullptr
```

Referenced by gpusat::Solver::cleanTree(), gpusat::Solver::combineTree(), main(), gpusat::printtreeType(), gpusat::solJson(), gpusat::solutiontable(), gpusat::Solver::solveIntroduceForget(), gpusat::Solver::solveJoin(), and gpusat::Solver::solveProblem().

5.15.2.2 maxId

```
cl_long gpusat::treeType::maxId = 0
```

Referenced by gpusat::Solver::cleanTree(), gpusat::Solver::combineTree(), main(), gpusat::printtreeType(), gpusat::Solver::solveIntroduceForget(), gpusat::Solver::solveJoin(), and gpusat::Solver::solveProblem().

5.15.2.3 minId

```
cl_long gpusat::treeType::minId = 0
```

Referenced by gpusat::Solver::cleanTree(), gpusat::Solver::combineTree(), main(), gpusat::printtreeType(), gpusat::solJson(), gpusat::solutiontable(), gpusat::Solver::solveIntroduceForget(), gpusat::Solver::solveJoin(), and gpusat::Solver::solveProblem().

5.15.2.4 numSolutions

```
cl_long gpusat::treeType::numSolutions = 0
```

Referenced by gpusat::Solver::cleanTree(), gpusat::Solver::combineTree(), gpusat::printtreeType(), gpusat::↩ Solver::solveIntroduceForget(), gpusat::Solver::solveJoin(), and gpusat::Solver::solveProblem().

5.15.2.5 size

```
cl_long gpusat::treeType::size = 0
```

Referenced by gpusat::Solver::cleanTree(), gpusat::Solver::combineTree(), gpusat::printtreeType(), gpusat::↩ Solver::solveIntroduceForget(), gpusat::Solver::solveJoin(), and gpusat::Solver::solveProblem().

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

- include/types.h

5.16 gpusat::Visualization Class Reference

```
#include <visualization.h>
```

Public Member Functions

- Visualization (std::string filename)
- Json::Value getClausesJson ()
- std::string getFilename ()
- Json::Value getTdTimeline ()
- Json::Value getTreeDecJson ()
- Json::StreamWriterBuilder ∗ getWriterBuilder ()
- bool isEnabled ()
- void tdTimelineAppend (std::vector< cl_long > bag_ids)
- void tdTimelineAppend (std::vector< cl_long > bag_ids, TableLines tablelines, std::string const toplabel="", std::string const bottomlabel="", bool transpose=true)
- void testJson ()
- void visuClauses (satformulaType ∗sat)
- void visuTreeDec (treedecType ∗treeDec)
- void writeJsonFile (bool append=false)
- void writeJsonToStdout (Json::StreamWriter::Factory const &factory, Json::Value const &value, Json::O↩ Stream ∗sout=&std::cout)
- void writeJsonToStdout (Json::Value const &value, Json::OStream ∗sout=&std::cout)

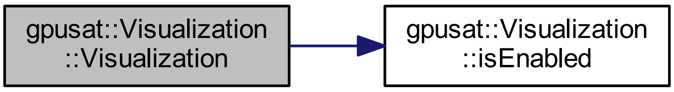
5.16.1 Constructor & Destructor Documentation

5.16.1.1 Visualization()

```
gpusat::Visualization::Visualization (
            std::string filename )  [inline]
```

References isEnabled().

Here is the call graph for this function:


gpusat::Visualization ::Visualization
gpusat::Visualization ::isEnabled

5.16.2 Member Function Documentation

5.16.2.1 getClausesJson()

```
Json::Value gpusat::Visualization::getClausesJson ( )  [inline]
```

5.16.2.2 getFilename()

```
std::string gpusat::Visualization::getFilename ( )  [inline]
```

5.16.2.3 getTdTimeline()

```
Json::Value gpusat::Visualization::getTdTimeline ( )  [inline]
```

5.16.2.4 getTreeDecJson()

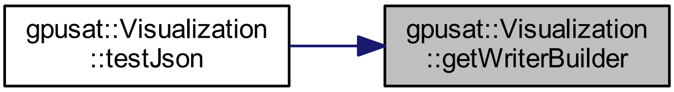
```
Json::Value gpusat::Visualization::getTreeDecJson ( )  [inline]
```

5.16.2.5 getWriterBuilder()

```
Json::StreamWriterBuilder * gpusat::Visualization::getWriterBuilder ( )
```

Referenced by testJson().

Here is the caller graph for this function:

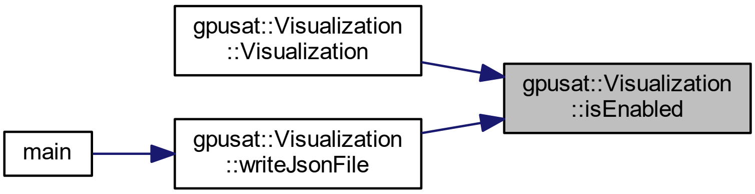

gpusat::Visualization ::testJson
gpusat::Visualization ::getWriterBuilder

5.16.2.6 isEnabled()

```
bool gpusat::Visualization::isEnabled ( )  [inline]
```

Referenced by Visualization(), and writeJsonFile().

Here is the caller graph for this function:



5.16.2.7 tdTimelineAppend() [1/2]

```
void gpusat::Visualization::tdTimelineAppend (
            std::vector< cl_long > bag_ids )
```

Only append those bags as one array. Might even be only one bag. The behaviour with zero length is not implemented (yet).

References LOGGER2.

5.16.2.8 tdTimelineAppend() [2/2]

```
void gpusat::Visualization::tdTimelineAppend (
            std::vector< cl_long > bag_ids,
            TableLines tablelines,
            std::string const toplabel = "",
            std::string const bottomlabel = "",
            bool transpose = true )
```

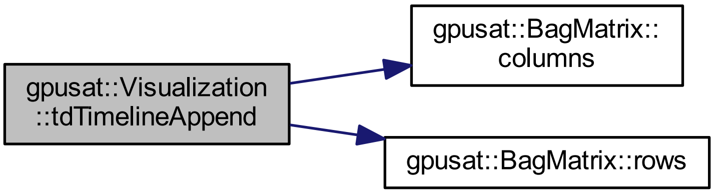
bag_ids: asserts not empty

- if one: one (IF) solution found
- more: join solution found tablelines: all lines for the table toplabel: string to label the whole solution bottomlabel: string to summarize the solution transpose: whether the tablelines are rowfirst (true) or not.

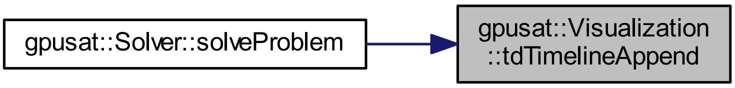
References gpusat::BagMatrix::columns(), gpusat::TableLines::headline, LOGGER2, gpusat::BagMatrix::rows(), gpusat::TableLines::solutions, and gpusat::TableLines::totalSol.

Referenced by gpusat::Solver::solveProblem().

Here is the call graph for this function:


gpusat::Visualization ::tdTimelineAppend
gpusat::BagMatrix:: columns
gpusat::BagMatrix::rows

Here is the caller graph for this function:

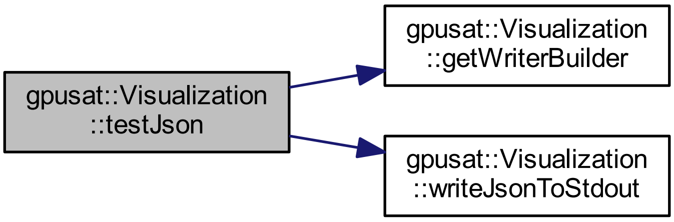

gpusat::Solver::solveProblem
gpusat::Visualization ::tdTimelineAppend

5.16.2.9 testJson()

```
void gpusat::Visualization::testJson ( )
```

References getWriterBuilder(), and writeJsonToStdout().

Here is the call graph for this function:


gpusat::Visualization ::testJson
gpusat::Visualization ::getWriterBuilder
gpusat::Visualization ::writeJsonToStdout

5.16.2.10 visuClauses()

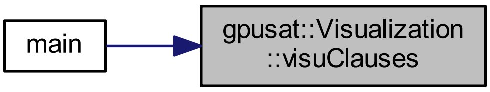
```
void gpusat::Visualization::visuClauses (
            satformulaType * sat )
```

Save the clauses into the Visualization::clausesJson. Form: List[Dict{INT: List[Int]}] Error if: sat formula is null or empty.

References gpusat::satformulaType::clauses.

Referenced by main().

Here is the caller graph for this function:



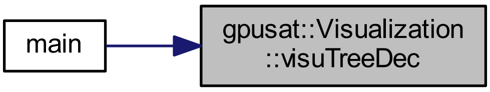
5.16.2.11 visuTreeDec()

```
void gpusat::Visualization::visuTreeDec (
            treedecType * treeDec )
```

References gpusat::treedecType::bags, gpusat::bagType::edges, gpusat::bagType::id, and gpusat::treedecType↩ ::numVars.

Referenced by main().

Here is the caller graph for this function:



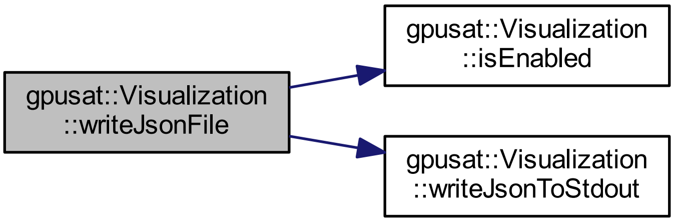
5.16.2.12 writeJsonFile()

```
void gpusat::Visualization::writeJsonFile (
            bool append = false )
```

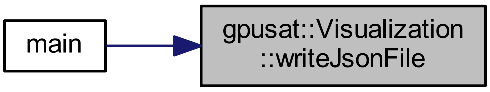
References isEnabled(), LOGGER, and writeJsonToStdout().

Referenced by main().

Here is the call graph for this function:


gpusat::Visualization ::writeJsonFile
gpusat::Visualization ::isEnabled
gpusat::Visualization ::writeJsonToStdout

Here is the caller graph for this function:

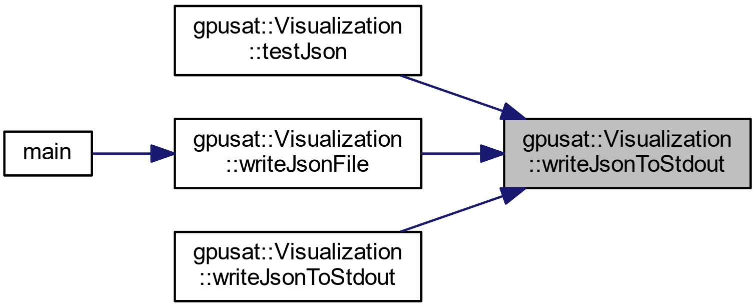


5.16.2.13 writeJsonToStdout() [1/2]

```
void gpusat::Visualization::writeJsonToStdout (
            Json::StreamWriter::Factory const & factory,
            Json::Value const & value,
            Json::OStream * sout = &std::cout )
```

Referenced by testJson(), writeJsonFile(), and writeJsonToStdout().

Here is the caller graph for this function:

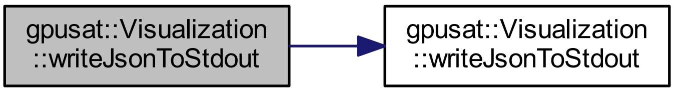


5.16.2.14 writeJsonToStdout() [2/2]

```
void gpusat::Visualization::writeJsonToStdout (
            Json::Value const & value,
            Json::OStream * sout = &std::cout )
```

References writeJsonToStdout().

Here is the call graph for this function:



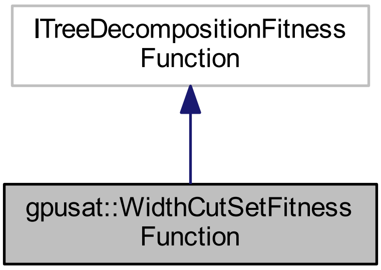
The documentation for this class was generated from the following files:

- include/visualization.h
- src/visualization.cpp

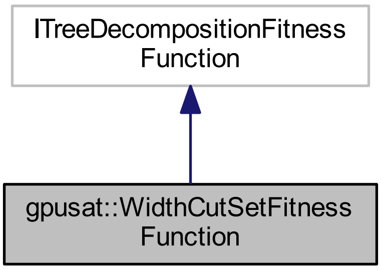
5.17 gpusat::WidthCutSetFitnessFunction Class Reference

```
#include <WidthCutSetFitnessFunction.h>
```

Inheritance diagram for gpusat::WidthCutSetFitnessFunction:


ITreeDecompositionFitness Function
gpusat::WidthCutSetFitness Function

Collaboration diagram for gpusat::WidthCutSetFitnessFunction:


ITreeDecompositionFitness Function
gpusat::WidthCutSetFitness Function

Public Member Functions

- WidthCutSetFitnessFunction ()=default
- ∼WidthCutSetFitnessFunction ()=default
- WidthCutSetFitnessFunction ∗ clone (void) const override
- htd::FitnessEvaluation ∗ fitness (const htd::IMultiHypergraph &graph, const htd::ITreeDecomposition &decomposition) const override
- double getMaxCutSetSize (const htd::ITreeDecomposition &decomposition) const

5.17.1 Detailed Description

Fitness function that first minimizes the width and then the cut set size.

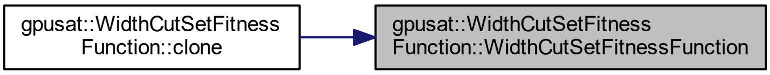
5.17.2 Constructor & Destructor Documentation

5.17.2.1 WidthCutSetFitnessFunction()

```
gpusat::WidthCutSetFitnessFunction::WidthCutSetFitnessFunction ( )  [default]
```

Referenced by clone().

Here is the caller graph for this function:


gpusat::WidthCutSetFitness Function::clone
gpusat::WidthCutSetFitness Function::WidthCutSetFitnessFunction

5.17.2.2 ∼WidthCutSetFitnessFunction()

```
gpusat::WidthCutSetFitnessFunction::∼WidthCutSetFitnessFunction ( )  [default]
```

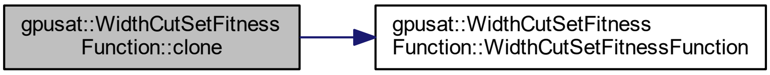
5.17.3 Member Function Documentation

5.17.3.1 clone()

```
WidthCutSetFitnessFunction ∗ gpusat::WidthCutSetFitnessFunction::clone (
            void  ) const  [override]
```

References WidthCutSetFitnessFunction().

Here is the call graph for this function:

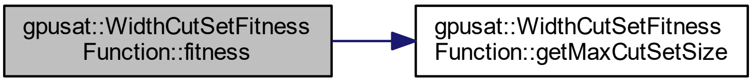

gpusat::WidthCutSetFitness Function::clone
gpusat::WidthCutSetFitness Function::WidthCutSetFitnessFunction

5.17.3.2 fitness()

```
htd::FitnessEvaluation * gpusat::WidthCutSetFitnessFunction::fitness (
            const htd::IMultiHypergraph & graph,
            const htd::ITreeDecomposition & decomposition ) const  [override]
```

References getMaxCutSetSize().

Here is the call graph for this function:

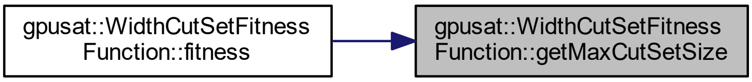

gpusat::WidthCutSetFitness
Function::fitness
gpusat::WidthCutSetFitness
Function::getMaxCutSetSize

5.17.3.3 getMaxCutSetSize()

```
double gpusat::WidthCutSetFitnessFunction::getMaxCutSetSize (
            const htd::ITreeDecomposition & decomposition ) const
```

Referenced by fitness().

Here is the caller graph for this function:


gpusat::WidthCutSetFitness
Function::fitness
gpusat::WidthCutSetFitness
Function::getMaxCutSetSize

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

- include/FitnessFunctions/WidthCutSetFitnessFunction.h
- src/FitnessFunctions/WidthCutSetFitnessFunction.cpp

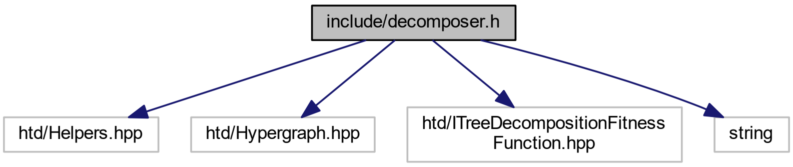
Chapter 6

File Documentation

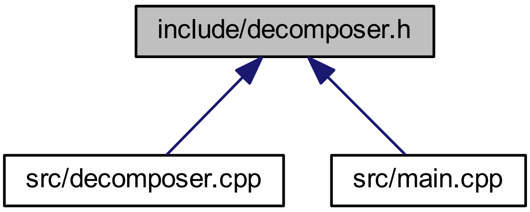
6.1 include/decomposer.h File Reference

```
#include <htd/Helpers.hpp>
#include <htd/Hypergraph.hpp>
#include <htd/ITreeDecompositionFitnessFunction.hpp>
#include <string>
```

Include dependency graph for decomposer.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gpusat::Decomposer

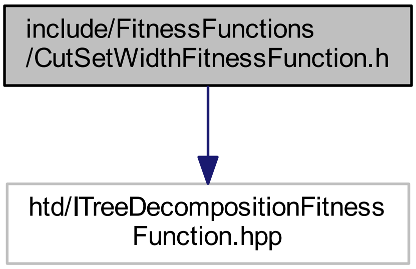
Namespaces

- gpusat

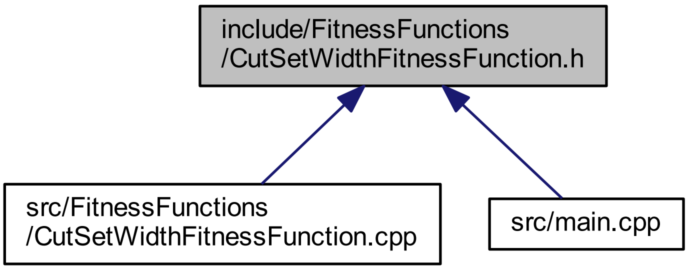
6.2 include/FitnessFunctions/CutSetWidthFitnessFunction.h File Reference

```
#include <htd/ITreeDecompositionFitnessFunction.hpp>
```

Include dependency graph for CutSetWidthFitnessFunction.h:



This graph shows which files directly or indirectly include this file:


include/FitnessFunctions
/CutSetWidthFitnessFunction.h
src/FitnessFunctions
/CutSetWidthFitnessFunction.cpp
src/main.cpp

Classes

- class gpusat::CutSetWidthFitnessFunction

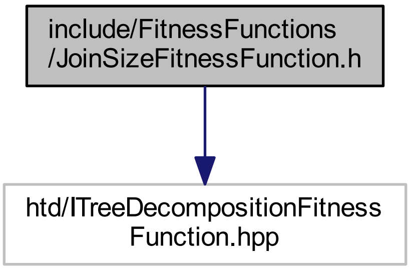
Namespaces

- gpusat

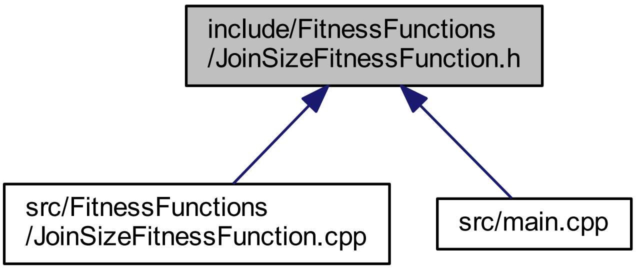
6.3 include/FitnessFunctions/JoinSizeFitnessFunction.h File Reference

```
#include <htd/ITreeDecompositionFitnessFunction.hpp>
```

Include dependency graph for JoinSizeFitnessFunction.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gpusat::JoinSizeFitnessFunction

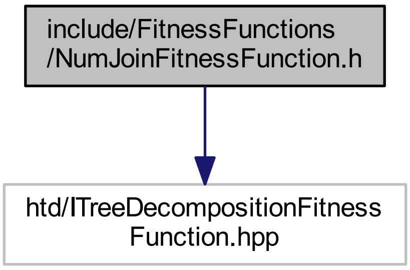
Namespaces

- gpusat

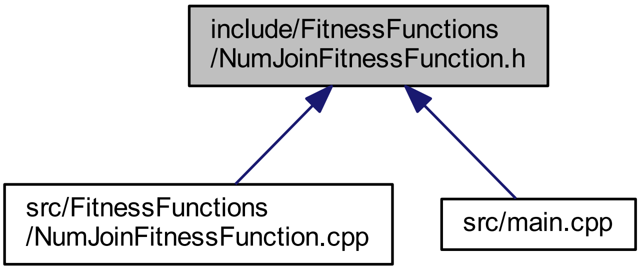
6.4 include/FitnessFunctions/NumJoinFitnessFunction.h File Reference

```
#include <htd/ITreeDecompositionFitnessFunction.hpp>
```

Include dependency graph for NumJoinFitnessFunction.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gpusat::NumJoinFitnessFunction

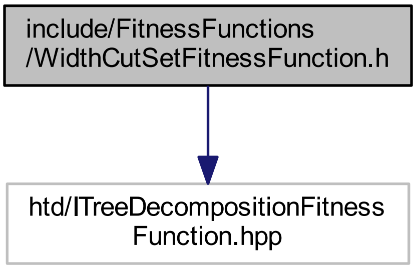
Namespaces

- gpusat

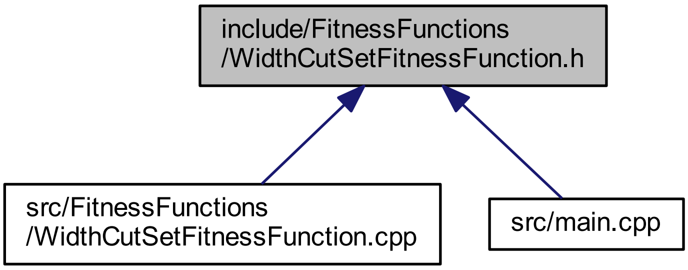
6.5 include/FitnessFunctions/WidthCutSetFitnessFunction.h File Reference

```
#include <htd/ITreeDecompositionFitnessFunction.hpp>
```

Include dependency graph for WidthCutSetFitnessFunction.h:



This graph shows which files directly or indirectly include this file:


include/FitnessFunctions
/WidthCutSetFitnessFunction.h
src/FitnessFunctions
/WidthCutSetFitnessFunction.cpp
src/main.cpp

Classes

- class gpusat::WidthCutSetFitnessFunction

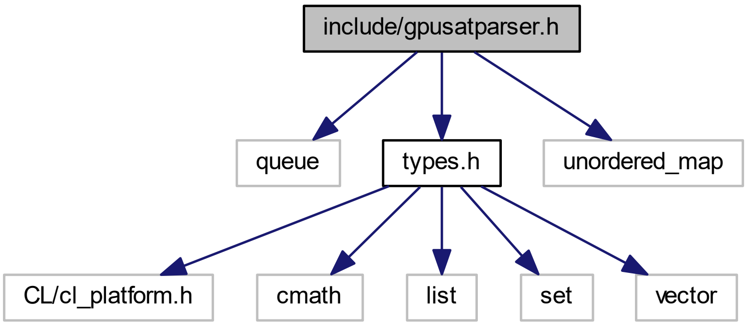
Namespaces

- gpusat

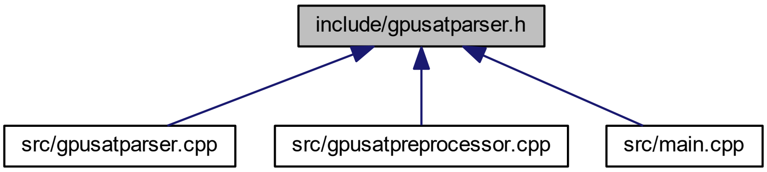
6.6 include/gpusatparser.h File Reference

```
#include <queue>
#include <types.h>
#include <unordered_map>
```

Include dependency graph for gpusatparser.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gpusat::CNFParser
- class gpusat::TDParser

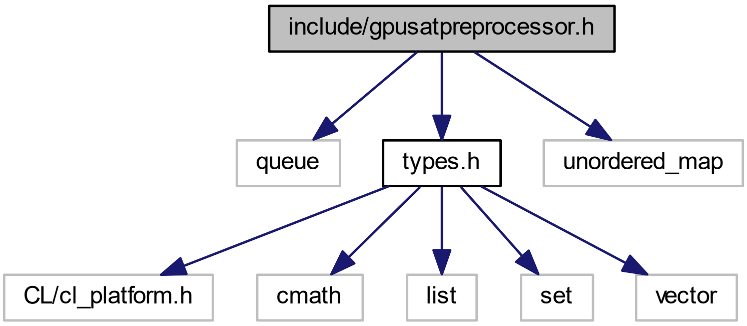
Namespaces

- gpusat

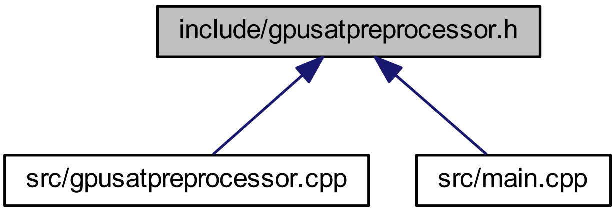
6.7 include/gpusatpreprocessor.h File Reference

```
#include <queue>
#include <types.h>
#include <unordered_map>
```

Include dependency graph for gpusatpreprocessor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gpusat::Preprocessor

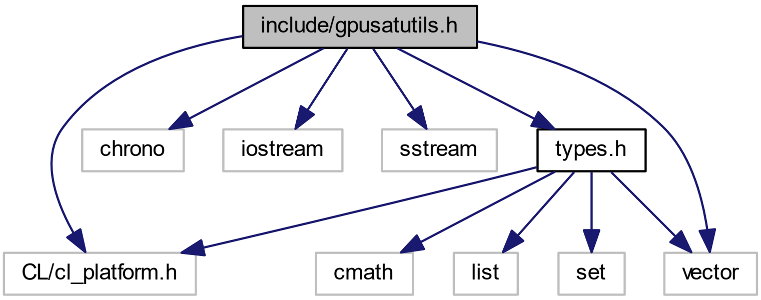
Namespaces

- gpusat

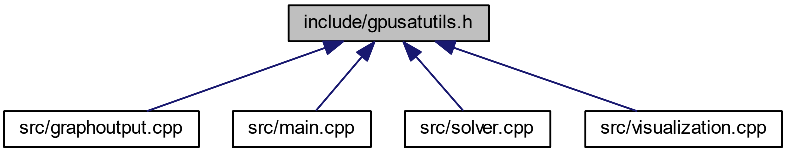
6.8 include/gpusatutils.h File Reference

```
#include <CL/cl_platform.h>
#include <chrono>
#include <iostream>
#include <sstream>
#include <types.h>
#include <vector>
```

Include dependency graph for gpusatutils.h:



This graph shows which files directly or indirectly include this file:



Namespaces

- gpusat

Functions

- cl_double gpusat::getCount (cl_long id, cl_long ∗tree, cl_long numVars)
- long long int gpusat::getTime ()
- std::ostream & gpusat::operator<< (std::ostream &os, const dataStructure ds)
- std::ostream & gpusat::operator<< (std::ostream &os, const std::vector< bagType ∗ > vec)
- std::ostream & gpusat::operator<< (std::ostream &os, const std::vector< cl_long > vec)
- void gpusat::printbagType (bagType ∗bag, std::ostream &stream, int depth=0)

  print information for a bag in the tree decomposition

- void gpusat::printtreedecType (treedecType ∗dec, std::ostream &stream)

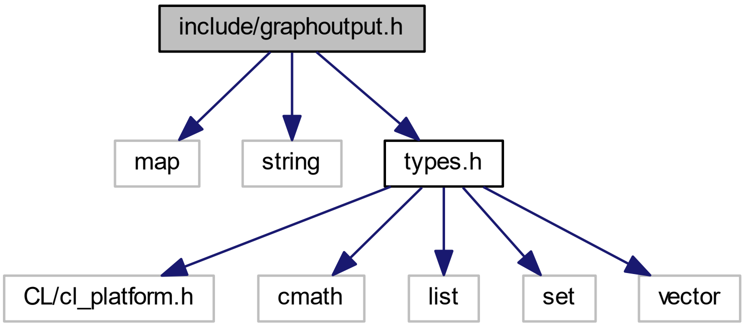
    print a tree decomposition
- void gpusat::printtreeType (treeType ∗tree, std::ostream &stream, size_t size, int depth=0)
- std::string gpusat::solutiontable (bagType node, dataStructure solutionType)

    Generate a formatted stringoutput for a solved node with solutions.

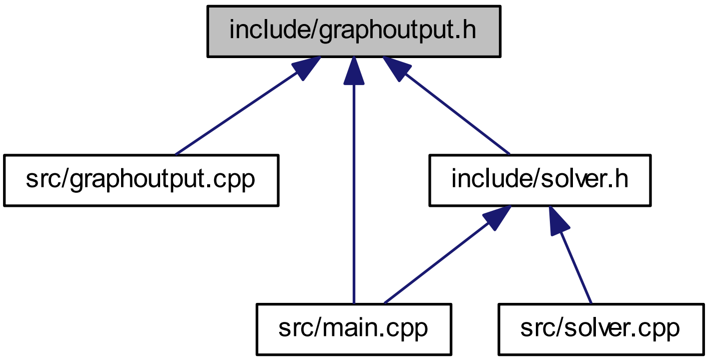
6.9 include/graphoutput.h File Reference

```
#include <map>
#include <string>
#include <types.h>
```

Include dependency graph for graphoutput.h:



This graph shows which files directly or indirectly include this file:



Classes

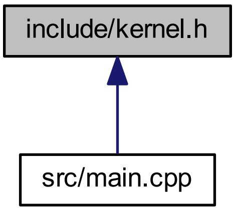
- class gpusat::Graphoutput

Namespaces

- gpusat

6.10 include/kernel.h File Reference

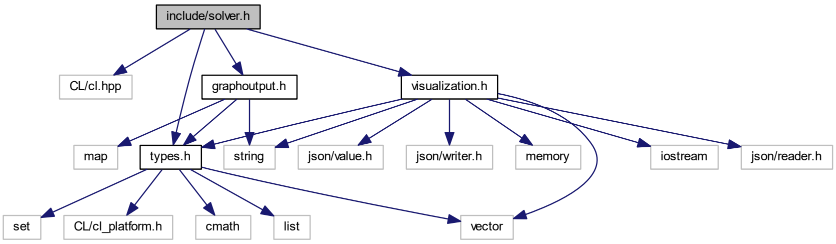
This graph shows which files directly or indirectly include this file:



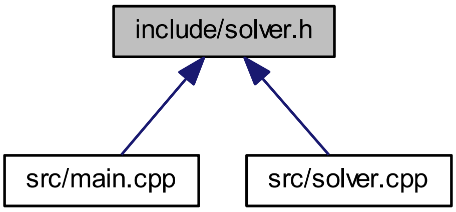
6.11 include/solver.h File Reference

```
#include <CL/cl.hpp>
#include <graphoutput.h>
#include <types.h>
#include <visualization.h>
```

Include dependency graph for solver.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gpusat::Solver

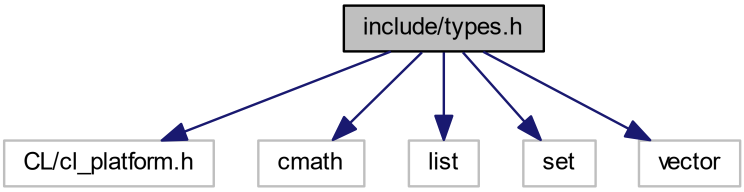
Namespaces

- gpusat

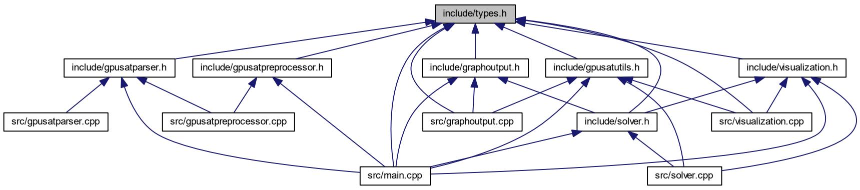
6.12 include/types.h File Reference

```
#include <CL/cl_platform.h>
#include <cmath>
#include <list>
#include <set>
#include <vector>
```

Include dependency graph for types.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct gpusat::bagType

  type for a bag in the tree decomposition
- struct gpusat::satformulaType

  type for saving the sat formula
- struct gpusat::treedecType

  type for saving a tree decomposition
- struct gpusat::treeType

  tree type for storing the models

Namespaces

- gpusat

Enumerations

- enum gpusat::dataStructure { gpusat::dataStructure::ARRAY, gpusat::dataStructure::TREE }
- enum gpusat::nodeType { gpusat::nodeType::JOIN, gpusat::nodeType::INTRODUCEFORGET }

Functions

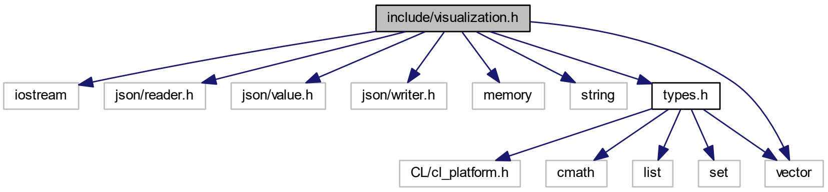
- bool gpusat::compTreedType (const bagType ∗a, const bagType ∗b)
- bool gpusat::compVars (const cl_long &a, const cl_long &b)

6.13 include/visualization.h File Reference

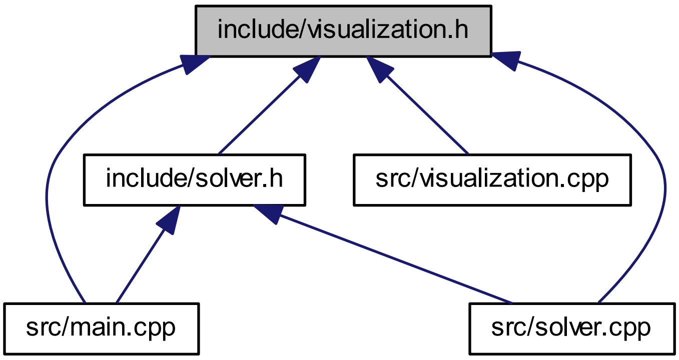
```
#include <iostream>
#include <json/reader.h>
#include <json/value.h>
#include <json/writer.h>
#include <memory>
#include <string>
#include <types.h>
```

`#include <vector>`

Include dependency graph for visualization.h:



This graph shows which files directly or indirectly include this file:



Classes

- class gpusat::BagMatrix
- struct gpusat::TableLines
- class gpusat::Visualization

Namespaces

- gpusat

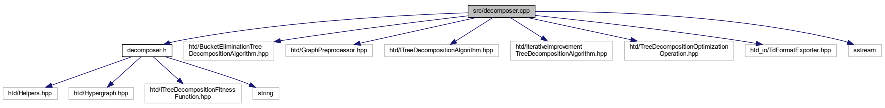
Functions

- TableLines gpusat::solJson (bagType node, dataStructure solutionType)

6.14 src/decomposer.cpp File Reference

```
#include <decomposer.h>
#include <htd/BucketEliminationTreeDecompositionAlgorithm.hpp>
#include <htd/GraphPreprocessor.hpp>
#include <htd/ITreeDecompositionAlgorithm.hpp>
#include <htd/IterativeImprovementTreeDecompositionAlgorithm.hpp>
#include <htd/TreeDecompositionOptimizationOperation.hpp>
#include <htd_io/TdFormatExporter.hpp>
#include <sstream>
```

Include dependency graph for decomposer.cpp:


src/decomposer.cpp
decomposer.h
htd/BucketEliminationTree DecompositionAlgorithm.hpp
htd/GraphPreprocessor.hpp
htd/ITreeDecompositionAlgorithm.hpp
htd/IterativeImprovement TreeDecompositionAlgorithm.hpp
htd/TreeDecompositionOptimization Operation.hpp
htd_io/TdFormatExporter.hpp
sstream
htd/Helpers.hpp
htd/Hypergraph.hpp
htd/ITreeDecompositionFitness Function.hpp
string

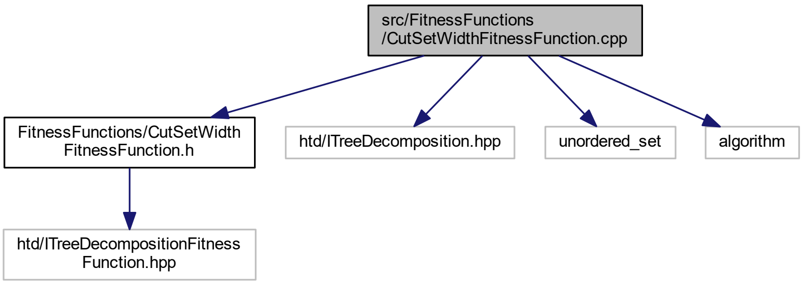
Namespaces

- gpusat

6.15 src/FitnessFunctions/CutSetWidthFitnessFunction.cpp File Reference

```
#include <FitnessFunctions/CutSetWidthFitnessFunction.h>
#include <htd/ITreeDecomposition.hpp>
#include <unordered_set>
#include <algorithm>
```

Include dependency graph for CutSetWidthFitnessFunction.cpp:


src/FitnessFunctions /CutSetWidthFitnessFunction.cpp
FitnessFunctions/CutSetWidth FitnessFunction.h
htd/ITreeDecomposition.hpp
unordered_set
algorithm
htd/ITreeDecompositionFitness Function.hpp

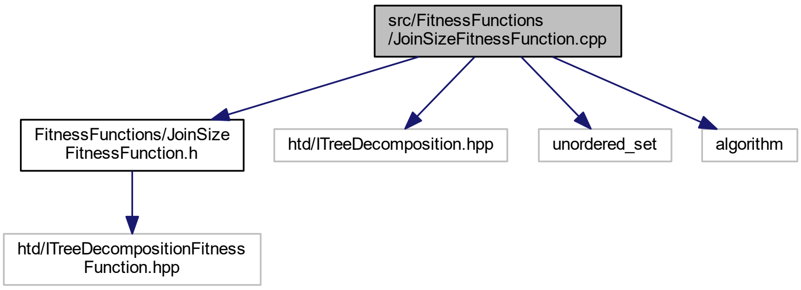
Namespaces

- gpusat

6.16 src/FitnessFunctions/JoinSizeFitnessFunction.cpp File Reference

```
#include <FitnessFunctions/JoinSizeFitnessFunction.h>
#include <htd/ITreeDecomposition.hpp>
#include <unordered_set>
#include <algorithm>
```

Include dependency graph for JoinSizeFitnessFunction.cpp:


src/FitnessFunctions
/JoinSizeFitnessFunction.cpp
FitnessFunctions/JoinSize
FitnessFunction.h
htd/ITreeDecomposition.hpp
unordered_set
algorithm
htd/ITreeDecompositionFitness
Function.hpp

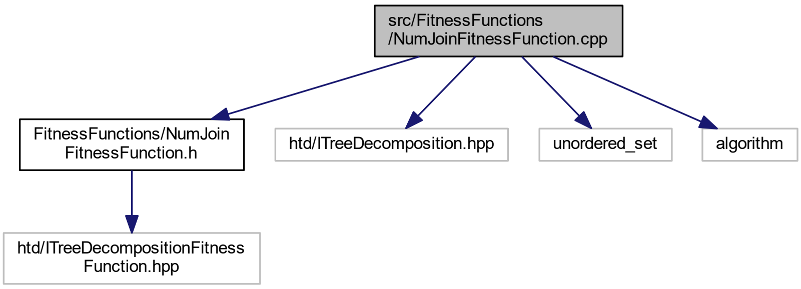
Namespaces

- gpusat

6.17 src/FitnessFunctions/NumJoinFitnessFunction.cpp File Reference

```
#include <FitnessFunctions/NumJoinFitnessFunction.h>
#include <htd/ITreeDecomposition.hpp>
#include <unordered_set>
#include <algorithm>
```

Include dependency graph for NumJoinFitnessFunction.cpp:



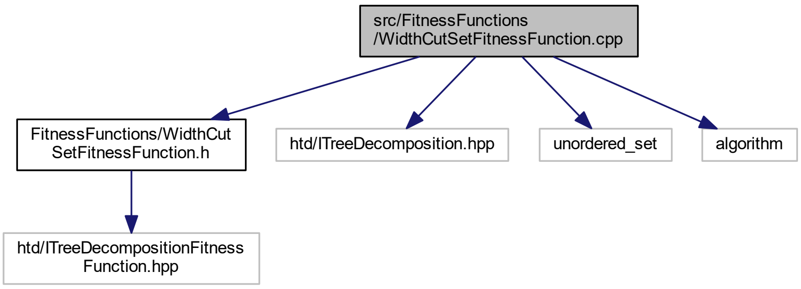
Namespaces

- gpusat

6.18 src/FitnessFunctions/WidthCutSetFitnessFunction.cpp File Reference

```
#include <FitnessFunctions/WidthCutSetFitnessFunction.h>
#include <htd/ITreeDecomposition.hpp>
#include <unordered_set>
#include <algorithm>
```

Include dependency graph for WidthCutSetFitnessFunction.cpp:


src/FitnessFunctions/WidthCutSetFitnessFunction.cpp
FitnessFunctions/WidthCutSetFitnessFunction.h
htd/ITreeDecomposition.hpp
unordered_set
algorithm
htd/ITreeDecompositionFitnessFunction.hpp

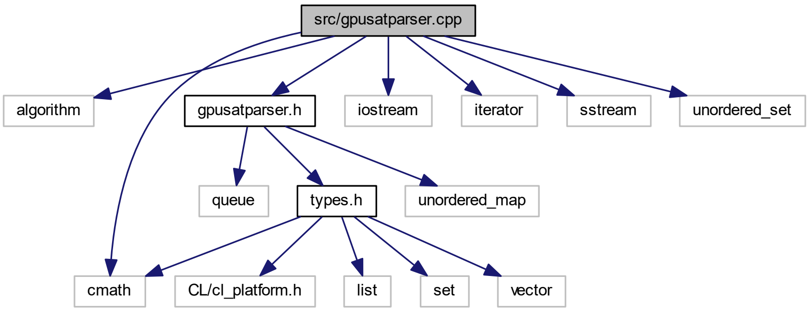
Namespaces

- gpusat

6.19 src/gpusatparser.cpp File Reference

```
#include <algorithm>
#include <cmath>
#include <gpusatparser.h>
#include <iostream>
#include <iterator>
#include <sstream>
#include <unordered_set>
```

Include dependency graph for gpusatparser.cpp:



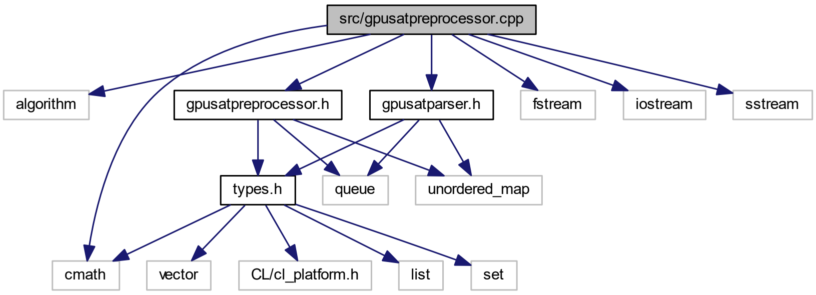
Namespaces

- gpusat

6.20 src/gpusatpreprocessor.cpp File Reference

```
#include <algorithm>
#include <cmath>
#include <fstream>
#include <gpusatparser.h>
#include <gpusatpreprocessor.h>
#include <iostream>
#include <sstream>
```

Include dependency graph for gpusatpreprocessor.cpp:



Namespaces

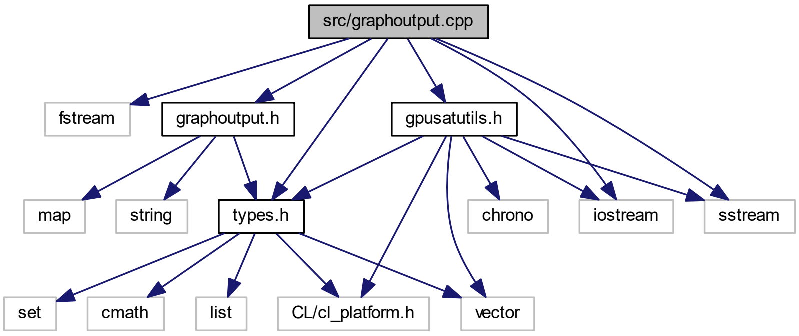
- gpusat

6.21 src/graphoutput.cpp File Reference

```
#include <fstream>
#include <gpusatutils.h>
#include <graphoutput.h>
#include <iostream>
#include <sstream>
```

```
#include <types.h>
```

Include dependency graph for graphoutput.cpp:



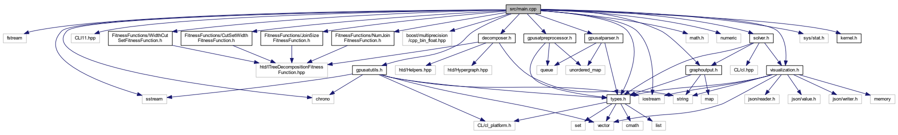
Namespaces

- gpusat

6.22 src/main.cpp File Reference

```
#include <fstream>
#include <iostream>
#include <sstream>
#include <CLI11.hpp>
#include <FitnessFunctions/CutSetWidthFitnessFunction.h>
#include <FitnessFunctions/JoinSizeFitnessFunction.h>
#include <FitnessFunctions/NumJoinFitnessFunction.h>
#include <FitnessFunctions/WidthCutSetFitnessFunction.h>
#include <boost/multiprecision/cpp_bin_float.hpp>
#include <chrono>
#include <decomposer.h>
#include <gpusatparser.h>
#include <gpusatpreprocessor.h>
#include <gpusatutils.h>
#include <graphoutput.h>
#include <math.h>
#include <numeric>
#include <solver.h>
#include <sys/stat.h>
#include <types.h>
#include <visualization.h>
#include <kernel.h>
```

Include dependency graph for main.cpp:



Macros

- #define __CL_ENABLE_EXCEPTIONS
- #define NELEMS(x) (sizeof(x) / sizeof((x)[0]))

Functions

- void buildKernel (cl::Context &context, std::vector< cl::Device > &devices, cl::CommandQueue &queue, cl↩ ::Program &program, cl_long &memorySize, cl_long &maxMemoryBuffer, bool nvidia, bool amd, bool cpu, long &combineWidth)
- void device_query ()
- int main (int argc, char ∗argv[ ])
- void PrintDeviceInfo (cl_device_id device)

Variables

- const char ∗const attributeNames [ ]
- const cl_platform_info attributeTypes [5]
- std::string kernelStr

6.22.1 Macro Definition Documentation

6.22.1.1 __CL_ENABLE_EXCEPTIONS

```
#define __CL_ENABLE_EXCEPTIONS
```

6.22.1.2 NELEMS

```
#define NELEMS(
                x ) (sizeof(x) / sizeof((x)[0]))
```

6.22.2 Function Documentation

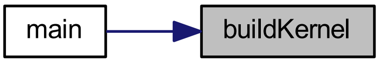
6.22.2.1 buildKernel()

```
void buildKernel (
            cl::Context & context,
            std::vector< cl::Device > & devices,
            cl::CommandQueue & queue,
            cl::Program & program,
            cl_long & memorySize,
            cl_long & maxMemoryBuffer,
            bool nvidia,
            bool amd,
            bool cpu,
            long & combineWidth )
```

References kernelStr.

Referenced by main().

Here is the caller graph for this function:



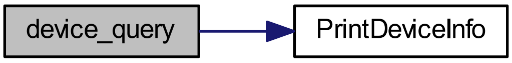
6.22.2.2 device_query()

```
void device_query ( )
```

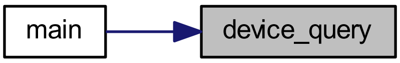
References attributeNames, attributeTypes, NELEMS, and PrintDeviceInfo().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:



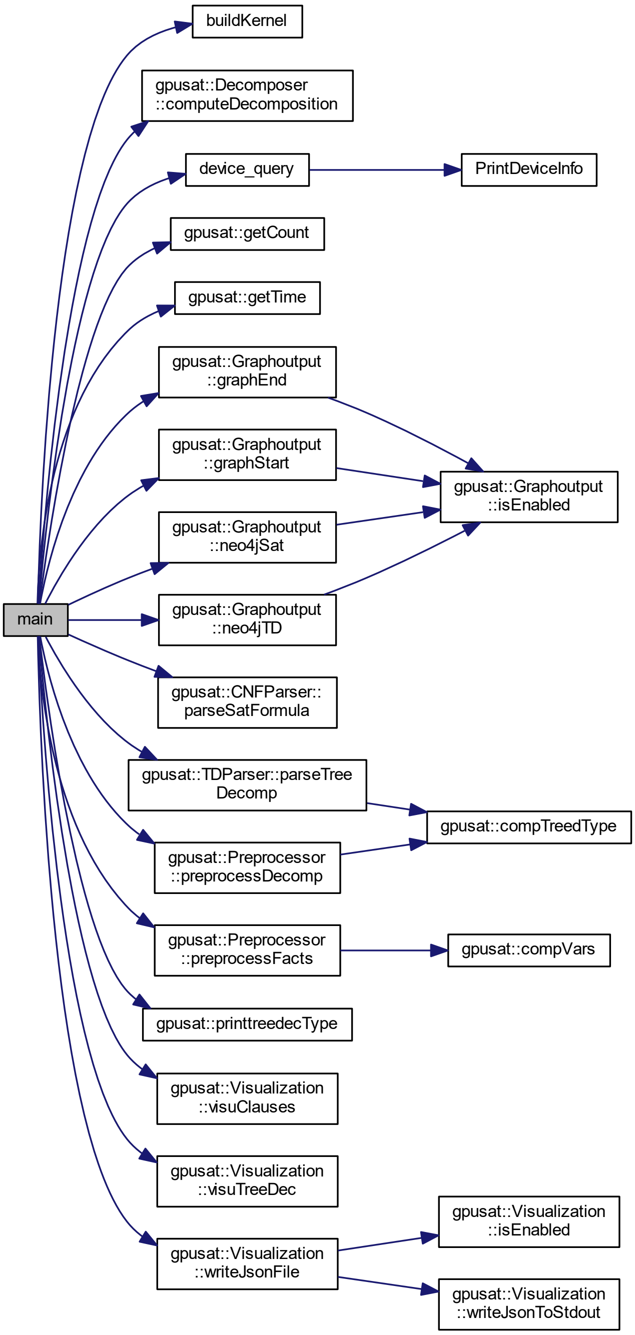
6.22.2.3 main()

```
int main (
            int argc,
            char * argv[] )
```

solution visualisation

References gpusat::ARRAY, gpusat::treedecType::bags, buildKernel(), gpusat::Decomposer::computeDecomposition(), gpusat::TDParser::defaultWeight, device_query(), gpusat::treeType::elements, gpusat::getCount(), gpusat::getTime(), gpusat::Graphoutput::graphEnd(), gpusat::Graphoutput::graphStart(), gpusat::INTRODUCEFORGET, kernelStr, gpusat::treeType::maxId, gpusat::Solver::maxTableSize, gpusat::treeType::minId, gpusat::Graphoutput::neo4jSat(), gpusat::Graphoutput::neo4jTD(), gpusat::Solver::numIntroduceForget, gpusat::Solver::numJoin, gpusat::CNFParser::parseSatFormula(), gpusat::TDParser::parseTreeDecomp(), gpusat::Preprocessor::preprocessDecomp(), gpusat::Preprocessor::preprocessFacts(), gpusat::printtreedecType(), gpusat::TREE, gpusat::satformulaType::unsat, gpusat::bagType::variables, gpusat::Visualization::visuClauses(), gpusat::Visualization::visuTreeDec(), gpusat::treedecType::width, and gpusat::Visualization::writeJsonFile().

Here is the call graph for this function:

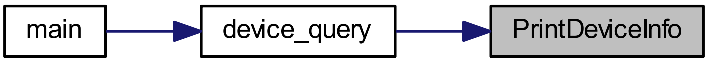

buildKernel
gpusat::Decomposer ::computeDecomposition
device_query
PrintDeviceInfo
gpusat::getCount
gpusat::getTime
gpusat::Graphoutput ::graphEnd
gpusat::Graphoutput ::graphStart
gpusat::Graphoutput ::isEnabled
gpusat::Graphoutput ::neo4jSat
main
gpusat::Graphoutput ::neo4jTD
gpusat::CNFParser:: parseSatFormula
gpusat::TDParser::parseTree Decomp
gpusat::compTreedType
gpusat::Preprocessor ::preprocessDecomp
gpusat::Preprocessor ::preprocessFacts
gpusat::compVars
gpusat::printtreedecType
gpusat::Visualization ::visuClauses
gpusat::Visualization ::visuTreeDec
gpusat::Visualization ::isEnabled
gpusat::Visualization ::writeJsonFile
gpusat::Visualization ::writeJsonToStdout

6.22.2.4 PrintDeviceInfo()

```
void PrintDeviceInfo (
            cl_device_id device )
```

Referenced by device_query().

Here is the caller graph for this function:



6.22.3 Variable Documentation

6.22.3.1 attributeNames

```
const char* const attributeNames[]
```

Initial value:

```
= {
    "CL_PLATFORM_NAME",
    "CL_PLATFORM_VENDOR",
    "CL_PLATFORM_VERSION",
    "CL_PLATFORM_PROFILE",
    "CL_PLATFORM_EXTENSIONS"}
```

Referenced by device_query().

6.22.3.2 attributeTypes

```
const cl_platform_info attributeTypes[5]
```

Initial value:

```
= {
    CL_PLATFORM_NAME,
    CL_PLATFORM_VENDOR,
    CL_PLATFORM_VERSION,
    CL_PLATFORM_PROFILE,
    CL_PLATFORM_EXTENSIONS}
```

Referenced by device_query().

6.22.3.3 kernelStr

```
std::string kernelStr
```

Initial value:

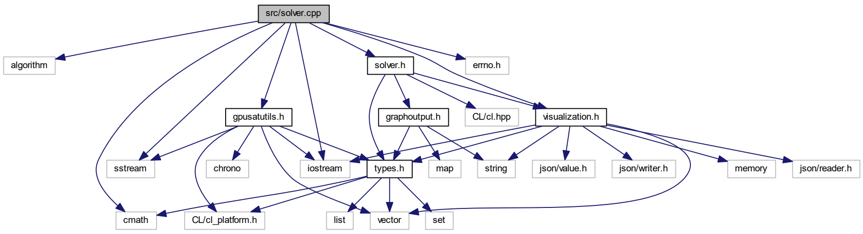
```
=
using namespace gpusat
```

Referenced by buildKernel(), and main().

6.23 src/solver.cpp File Reference

```
#include <algorithm>
#include <cmath>
#include <errno.h>
#include <gpusatutils.h>
#include <iostream>
#include <solver.h>
#include <sstream>
#include <visualization.h>
```

Include dependency graph for solver.cpp:



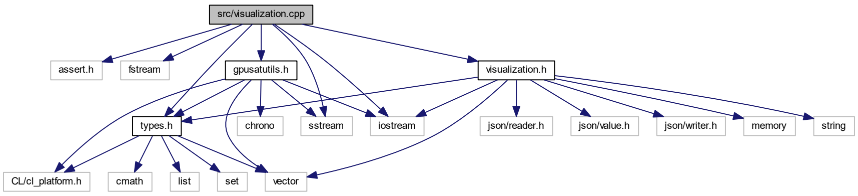
Namespaces

- gpusat

6.24 src/visualization.cpp File Reference

```
#include <assert.h>
#include <fstream>
#include <gpusatutils.h>
#include <iostream>
#include <sstream>
#include <types.h>
#include <visualization.h>
```

Include dependency graph for visualization.cpp:



Namespaces

- gpusat

Macros

- #define LOGGER(x) (std::cout << "LOGGING: " << x << std::endl)
- #define LOGGER2(x)

Functions

- TableLines gpusat::solJson (bagType node, dataStructure solutionType)

6.24.1 Macro Definition Documentation

6.24.1.1 LOGGER

```
#define LOGGER(
              x ) (std::cout << "LOGGING: " << x << std::endl)
```

6.24.1.2 LOGGER2

```
#define LOGGER2(
              x )
```

Index