

My Project

Generated by Doxygen 1.8.6

Mon Sep 28 2015 13:20:08

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Class Documentation	7
4.1	bounds Struct Reference	7
4.1.1	Constructor & Destructor Documentation	7
4.1.1.1	~bounds	7
4.1.2	Member Data Documentation	7
4.1.2.1	lb	7
4.1.2.2	type	8
4.1.2.3	ub	8
4.2	BP_Input Class Reference	8
4.2.1	Constructor & Destructor Documentation	8
4.2.1.1	BP_Input	8
4.2.1.2	~BP_Input	8
4.2.2	Member Function Documentation	8
4.2.2.1	getBterms	9
4.2.2.2	getDirection	9
4.2.2.3	getMatcoeff	9
4.2.2.4	getMatcoeff2	9
4.2.2.5	getNcons	9
4.2.2.6	getNvars	9
4.2.2.7	getScale	9
4.2.2.8	getVar	9
4.2.2.9	getVars	9
4.2.2.10	setDigits	9

4.2.3	Friends And Related Function Documentation	9
4.2.3.1	operator<<	9
4.2.4	Member Data Documentation	9
4.2.4.1	bterms	9
4.2.4.2	direction	9
4.2.4.3	matcoeff	9
4.2.4.4	matcoeff2	9
4.2.4.5	nbinsvars	9
4.2.4.6	ncons	9
4.2.4.7	nintvars	9
4.2.4.8	nvars	9
4.2.4.9	scale	9
4.2.4.10	vars	9
4.3	BP_Output Class Reference	9
4.3.1	Constructor & Destructor Documentation	10
4.3.1.1	BP_Output	10
4.3.2	Member Function Documentation	10
4.3.2.1	assign	10
4.3.2.2	assignment	10
4.3.2.3	operator=	10
4.3.3	Friends And Related Function Documentation	10
4.3.3.1	operator<<	10
4.3.3.2	operator>>	10
4.3.4	Member Data Documentation	11
4.3.4.1	in	11
4.3.4.2	varAssignment	11
4.4	BPSolver Class Reference	11
4.4.1	Constructor & Destructor Documentation	12
4.4.1.1	BPSolver	12
4.4.1.2	~BPSolver	12
4.4.2	Member Function Documentation	12
4.4.2.1	printCurrent	12
4.5	Clock Class Reference	12
4.5.1	Member Data Documentation	12
4.5.1.1	globalClock	12
4.6	Constraint Class Reference	12
4.6.1	Constructor & Destructor Documentation	14
4.6.1.1	Constraint	14
4.6.1.2	~Constraint	14
4.6.2	Member Function Documentation	14

4.6.2.1	getArgument	14
4.6.2.2	getCoefficients	14
4.6.2.3	getDeltaViolation	14
4.6.2.4	getDeltaViolationDegree	14
4.6.2.5	getInvariant	14
4.6.2.6	getNumberOfIntegerVariables	14
4.6.2.7	getScopeSize	14
4.6.2.8	getType	14
4.6.2.9	getVariables	14
4.6.2.10	getViolation	14
4.6.2.11	getViolationDegree	14
4.6.2.12	isOneway	14
4.6.2.13	isOneway	14
4.6.2.14	operator<	14
4.6.2.15	operator>	14
4.6.2.16	setDeltaViolation	14
4.6.2.17	setDeltaViolationDegree	14
4.6.2.18	setInvariant	14
4.6.2.19	setNumberOfIntegerVariables	14
4.6.2.20	testCons	14
4.6.2.21	testObj	15
4.6.2.22	updateViolation	15
4.6.2.23	updateViolationDegree	15
4.6.3	Member Data Documentation	15
4.6.3.1	arguments	15
4.6.3.2	coefficients	15
4.6.3.3	DeltaViolation	15
4.6.3.4	DeltaViolationDegree	15
4.6.3.5	invariant	15
4.6.3.6	numberOfIntegerVariables	15
4.6.3.7	oneway	15
4.6.3.8	priority	15
4.6.3.9	scopeSize	15
4.6.3.10	type	15
4.6.3.11	variables	15
4.6.3.12	Violation	15
4.6.3.13	ViolationDegree	15
4.7	ConstraintSorter Class Reference	15
4.7.1	Constructor & Destructor Documentation	16
4.7.1.1	ConstraintSorter	16

4.7.2	Member Function Documentation	16
4.7.2.1	operator()	16
4.8	elem Struct Reference	16
4.8.1	Constructor & Destructor Documentation	16
4.8.1.1	~elem	16
4.8.2	Member Data Documentation	16
4.8.2.1	coeff	16
4.8.2.2	index	16
4.9	FlipMove Class Reference	16
4.9.1	Constructor & Destructor Documentation	17
4.9.1.1	FlipMove	17
4.9.1.2	FlipMove	18
4.9.1.3	~FlipMove	18
4.9.2	Member Function Documentation	18
4.9.2.1	operator=	18
4.10	GecodeSolver Class Reference	18
4.10.1	Constructor & Destructor Documentation	19
4.10.1.1	GecodeSolver	19
4.10.1.2	~GecodeSolver	19
4.10.1.3	GecodeSolver	19
4.10.2	Member Function Documentation	19
4.10.2.1	branch	19
4.10.2.2	copy	19
4.10.2.3	createArray	19
4.10.2.4	createGecodeVariable	19
4.10.2.5	FindSolution	19
4.10.2.6	fixVariables	19
4.10.2.7	initialize	19
4.10.2.8	linear	19
4.10.2.9	postCovSol	19
4.10.2.10	print	19
4.10.2.11	print_stats	20
4.10.2.12	printSpaceStatus	20
4.10.2.13	SetValues	20
4.10.3	Member Data Documentation	20
4.10.3.1	binVars	20
4.10.3.2	IntVars	20
4.10.3.3	model	20
4.10.3.4	tmpVars	20
4.11	GeneralSolver Class Reference	20

4.11.1	Constructor & Destructor Documentation	21
4.11.1.1	GeneralSolver	21
4.11.1.2	~GeneralSolver	21
4.11.2	Member Function Documentation	21
4.11.2.1	canBeMadeOneway	21
4.11.2.2	createIntVar	21
4.11.2.3	createIntVars	21
4.11.2.4	getAllVariables	21
4.11.2.5	getInitialValue	21
4.11.2.6	initializeLS	21
4.11.2.7	InitialSolution	22
4.11.2.8	linear	22
4.11.2.9	makeOneway	22
4.11.2.10	operator=	22
4.11.2.11	optimizeSolution	22
4.11.2.12	print	22
4.11.2.13	print_stats	22
4.11.2.14	printCurrent	22
4.11.2.15	relax	22
4.11.2.16	simpleRelax	22
4.11.3	Friends And Related Function Documentation	22
4.11.3.1	Test	22
4.11.4	Member Data Documentation	22
4.11.4.1	GS	22
4.11.4.2	LS	22
4.11.4.3	model	22
4.11.4.4	st	22
4.12	IntegerVariable Class Reference	22
4.12.1	Constructor & Destructor Documentation	24
4.12.1.1	IntegerVariable	24
4.12.1.2	~IntegerVariable	24
4.12.2	Member Function Documentation	24
4.12.2.1	addToUpdate	24
4.12.2.2	addToUsedInConstraints	24
4.12.2.3	clearUpdateVector	24
4.12.2.4	getCurrentValue	24
4.12.2.5	getID	24
4.12.2.6	getLowerBound	24
4.12.2.7	getOneway	24
4.12.2.8	getUpdateVector	24

4.12.2.9	getUpperBound	24
4.12.2.10	getVariablePointer	24
4.12.2.11	isIntegerVariable	24
4.12.2.12	setCurrentValue	24
4.12.2.13	setDefinedBy	24
4.12.2.14	setVariablePointer	24
4.12.2.15	usedInConstraints	24
4.12.3	Friends And Related Function Documentation	24
4.12.3.1	GeneralSolver	24
4.12.4	Member Data Documentation	24
4.12.4.1	ArrayPointer	24
4.12.4.2	constraints	24
4.12.4.3	definedByCons	24
4.12.4.4	isDefined	24
4.12.4.5	isInteger	24
4.12.4.6	lowerBound	24
4.12.4.7	oneway	25
4.12.4.8	update	25
4.12.4.9	upperBound	25
4.12.4.10	usedIn	25
4.12.4.11	value	25
4.12.4.12	VariablePointer	25
4.12.4.13	vectorID	25
4.13	Invariant Class Reference	25
4.13.1	Constructor & Destructor Documentation	26
4.13.1.1	Invariant	26
4.13.1.2	Invariant	26
4.13.1.3	~Invariant	26
4.13.2	Member Function Documentation	26
4.13.2.1	addChange	26
4.13.2.2	addToUpdate	26
4.13.2.3	calculateDeltaValue	26
4.13.2.4	getCoefficients	27
4.13.2.5	getConstraintNumber	27
4.13.2.6	getCurrentValue	27
4.13.2.7	getDeltaValue	27
4.13.2.8	getPriority	27
4.13.2.9	getType	27
4.13.2.10	getUpdateVector	27
4.13.2.11	getUsedInObjective	27

4.13.2.12	getVariableID	27
4.13.2.13	getVariables	27
4.13.2.14	initialize	27
4.13.2.15	setUsedByConstraint	27
4.13.2.16	setUsedByObjective	27
4.13.2.17	test	27
4.13.2.18	updateValue	27
4.13.3	Friends And Related Function Documentation	27
4.13.3.1	GeneralSolver	27
4.13.4	Member Data Documentation	27
4.13.4.1	coefficients	27
4.13.4.2	constraintPriority	27
4.13.4.3	CurrentValue	27
4.13.4.4	DeltaValue	27
4.13.4.5	invariants	27
4.13.4.6	startValue	28
4.13.4.7	type	28
4.13.4.8	update	28
4.13.4.9	usedInConstraintNr	28
4.13.4.10	usedInObjectiveNr	28
4.13.4.11	variableID	28
4.13.4.12	VariablePointers	28
4.14	Linear Class Reference	28
4.14.1	Constructor & Destructor Documentation	29
4.14.1.1	Linear	29
4.14.1.2	~Linear	29
4.14.2	Member Function Documentation	29
4.14.2.1	setDeltaViolation	29
4.14.2.2	setDeltaViolationDegree	30
4.14.2.3	testCons	30
4.14.2.4	testObj	30
4.14.2.5	updateViolation	30
4.14.2.6	updateViolationDegree	30
4.14.3	Member Data Documentation	30
4.14.3.1	relation	30
4.14.3.2	rhs	30
4.15	LSSpace Class Reference	30
4.15.1	Constructor & Destructor Documentation	31
4.15.1.1	LSSpace	31
4.15.1.2	~LSSpace	31

4.15.2	Member Function Documentation	31
4.15.2.1	bestImprovement	31
4.15.2.2	optimizeSolution	31
4.15.2.3	printCurrent	31
4.15.2.4	simpleMove	31
4.15.3	Friends And Related Function Documentation	31
4.15.3.1	Test	31
4.15.4	Member Data Documentation	31
4.15.4.1	iterations	31
4.15.4.2	model	31
4.15.4.3	Violations	31
4.16	Model Class Reference	31
4.16.1	Constructor & Destructor Documentation	32
4.16.1.1	Model	32
4.16.1.2	Model	32
4.16.1.3	~Model	32
4.16.2	Member Function Documentation	32
4.16.2.1	addBinaryVariable	32
4.16.2.2	addIntegerVariable	32
4.16.2.3	getAllVariables	32
4.16.2.4	getConstraints	32
4.16.2.5	getConstraintsWithPriority	32
4.16.2.6	getIntegerVariables	32
4.16.2.7	getInvariants	33
4.16.2.8	getMask	33
4.16.2.9	getMaskAt	33
4.16.2.10	getNonFixedBinaryVariable	33
4.16.2.11	getNonFixedBinaryVariables	33
4.16.2.12	getObjectives	33
4.16.2.13	nonFixedVariables	33
4.16.2.14	updateIntegerVariable	33
4.16.3	Member Data Documentation	33
4.16.3.1	Constraints	33
4.16.3.2	initialValue	33
4.16.3.3	IntegerVariables	33
4.16.3.4	Invariants	33
4.16.3.5	mask	33
4.16.3.6	nonFixedBinaryVariables	33
4.16.3.7	original	33
4.17	Move Class Reference	34

4.17.1	Constructor & Destructor Documentation	35
4.17.1.1	Move	35
4.17.1.2	Move	35
4.17.1.3	Move	35
4.17.1.4	Move	35
4.17.1.5	~Move	35
4.17.1.6	Move	35
4.17.2	Member Function Documentation	35
4.17.2.1	copy	35
4.17.2.2	flip	35
4.17.2.3	operator=	35
4.17.3	Member Data Documentation	35
4.17.3.1	deltaValueFirst	35
4.17.3.2	deltaValueSecond	35
4.17.3.3	deltaValueThird	35
4.17.3.4	first	35
4.17.3.5	moveType	35
4.17.3.6	second	35
4.17.3.7	third	35
4.18	Multistop Class Reference	35
4.18.1	Constructor & Destructor Documentation	37
4.18.1.1	Multistop	37
4.18.1.2	Multistop	37
4.18.1.3	~Multistop	37
4.18.2	Member Function Documentation	37
4.18.2.1	stop	37
4.18.3	Member Data Documentation	37
4.18.3.1	called	37
4.18.3.2	fs	37
4.18.3.3	ns	37
4.18.3.4	ts	37
4.19	NeighborhoodExplorer Class Reference	37
4.19.1	Constructor & Destructor Documentation	38
4.19.1.1	NeighborhoodExplorer	38
4.19.1.2	NeighborhoodExplorer	38
4.19.1.3	~NeighborhoodExplorer	38
4.19.2	Member Function Documentation	38
4.19.2.1	bestImprovement	38
4.19.2.2	calculateDeltaChange	38
4.19.2.3	commitMove	38

4.19.2.4	makeMove	38
4.19.2.5	randomWalk	38
4.19.3	Member Data Documentation	38
4.19.3.1	model	38
4.20	Options Class Reference	38
4.20.1	Constructor & Destructor Documentation	39
4.20.1.1	Options	39
4.20.1.2	Options	39
4.20.1.3	~Options	39
4.20.2	Member Data Documentation	39
4.20.2.1	timelimit	39
4.21	Random Class Reference	39
4.21.1	Member Function Documentation	39
4.21.1.1	Double	39
4.21.1.2	Integer	39
4.21.1.3	Integer	39
4.21.1.4	Seed	39
4.21.2	Member Data Documentation	39
4.21.2.1	mt	39
4.21.2.2	seed	40
4.22	Constraint::SortGreater Struct Reference	40
4.22.1	Member Function Documentation	40
4.22.1.1	operator()	40
4.23	State Class Reference	40
4.23.1	Constructor & Destructor Documentation	41
4.23.1.1	State	41
4.23.1.2	State	41
4.23.1.3	~State	41
4.23.2	Member Function Documentation	41
4.23.2.1	getObjectiveValue	41
4.23.2.2	getSolution	41
4.23.2.3	getSolutionValue	41
4.23.2.4	initializeConstraints	41
4.23.2.5	initializeInvariants	41
4.23.2.6	initializeObjective	41
4.23.2.7	maskAt	41
4.23.2.8	recalculateAll	41
4.23.2.9	saveSolution	41
4.23.2.10	setSolution	41
4.23.2.11	shuffleMask	41

4.23.3	Member Data Documentation	41
4.23.3.1	mask	41
4.23.3.2	model	42
4.23.3.3	numberOfViolations	42
4.23.3.4	solution	42
4.23.3.5	solutionValue	42
4.24	Sum Class Reference	42
4.24.1	Constructor & Destructor Documentation	43
4.24.1.1	Sum	43
4.24.1.2	Sum	43
4.24.1.3	Sum	43
4.24.1.4	~Sum	43
4.24.2	Member Function Documentation	43
4.24.2.1	addChange	43
4.24.2.2	calculateDeltaValue	43
4.24.2.3	operator=	43
4.24.2.4	test	43
4.24.3	Member Data Documentation	43
4.24.3.1	VariableChange	43
4.25	ValueChangeMove Class Reference	44
4.25.1	Constructor & Destructor Documentation	44
4.25.1.1	ValueChangeMove	45
4.25.1.2	ValueChangeMove	45
4.25.1.3	~ValueChangeMove	45
4.25.2	Member Function Documentation	45
4.25.2.1	operator=	45
4.26	var Struct Reference	45
4.26.1	Constructor & Destructor Documentation	46
4.26.1.1	~var	46
4.26.2	Member Data Documentation	46
4.26.2.1	bin	46
4.26.2.2	kind	46
4.26.2.3	objcoeff	46
5	File Documentation	47
5.1	BP_Data.cpp File Reference	47
5.1.1	Function Documentation	47
5.1.1.1	operator<<	47
5.1.1.2	operator>>	47
5.2	BP_Data.hpp File Reference	47

5.2.1	Macro Definition Documentation	48
5.2.1.1	DEBUG	48
5.3	BPSolver.hpp File Reference	48
5.4	Clock.cpp File Reference	49
5.5	Clock.hpp File Reference	50
5.6	Constants.hpp File Reference	50
5.6.1	Macro Definition Documentation	52
5.6.1.1	debug	52
5.6.1.2	EQ	52
5.6.1.3	FLIP	52
5.6.1.4	GQ	52
5.6.1.5	GR	52
5.6.1.6	HARD	52
5.6.1.7	iniTrails	52
5.6.1.8	LE	52
5.6.1.9	LINEAR	52
5.6.1.10	LQ	52
5.6.1.11	maxIter	52
5.6.1.12	maxMoves	52
5.6.1.13	nonTerm	52
5.6.1.14	OBJ	52
5.6.1.15	PRINT	53
5.6.1.16	RANDOMSEED	53
5.6.1.17	SOFT	53
5.6.1.18	SUM	53
5.6.1.19	SWAP	53
5.6.1.20	tabulistLB	53
5.6.1.21	tabulistUB	53
5.6.1.22	VALUECHANGE	53
5.6.1.23	weight	53
5.6.2	Typedef Documentation	53
5.6.2.1	allConstraints	53
5.6.2.2	coefType	53
5.6.2.3	constraint	53
5.6.2.4	constraintContainer	53
5.6.2.5	invariant	53
5.6.2.6	InvariantContainer	53
5.6.2.7	updateType	53
5.6.2.8	updateVector	53
5.6.2.9	variableContainer	53

5.6.2.10 VariableInConstraints	53
5.7 Constraint.hpp File Reference	53
5.8 FlipMove.cpp File Reference	54
5.9 FlipMove.hpp File Reference	55
5.10 GecodeSolver.cpp File Reference	56
5.11 GecodeSolver.hpp File Reference	57
5.12 GeneralSolver.cpp File Reference	58
5.13 GeneralSolver.hpp File Reference	58
5.14 getRSS.hpp File Reference	60
5.14.1 Function Documentation	60
5.14.1.1 getCurrentRSS	60
5.14.1.2 getPeakRSS	60
5.15 IntegerVariable.cpp File Reference	60
5.16 IntegerVariable.hpp File Reference	60
5.17 Invariant.hpp File Reference	61
5.18 Linear.hpp File Reference	62
5.19 LSSpace.cpp File Reference	64
5.20 LSSpace.hpp File Reference	64
5.21 main.cpp File Reference	65
5.21.1 Function Documentation	66
5.21.1.1 main	66
5.22 Model.cpp File Reference	66
5.23 Model.hpp File Reference	67
5.24 Move.hpp File Reference	68
5.25 Multistop.cpp File Reference	70
5.26 Multistop.hpp File Reference	70
5.27 NeighborhoodExplorer.cpp File Reference	71
5.28 NeighborhoodExplorer.hpp File Reference	72
5.29 Options.cpp File Reference	74
5.30 Options.hpp File Reference	74
5.31 Random.cpp File Reference	75
5.32 Random.hpp File Reference	75
5.33 State.cpp File Reference	76
5.34 State.hpp File Reference	77
5.35 Sum.cpp File Reference	78
5.36 Sum.hpp File Reference	79
5.37 ValueChangeMove.cpp File Reference	81
5.38 ValueChangeMove.hpp File Reference	82

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

bounds	7
var	45
BP_Input	8
BP_Output	9
Clock	12
Constraint	12
Linear	28
ConstraintSorter	15
elem	16
GeneralSolver	20
BPSolver	11
IntegerVariable	22
Invariant	25
Sum	42
LSSpace	30
Model	31
Move	34
FlipMove	16
ValueChangeMove	44
NeighborhoodExplorer	37
Options	38
Random	39
Constraint::SortGreater	40
Space	
GecodeSolver	18
State	40
Stop	
Multistop	35

Chapter 2

Class Index

2.1 Class List

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

bounds	7
BP_Input	8
BP_Output	9
BPSolver	11
Clock	12
Constraint	12
ConstraintSorter	15
elem	16
FlipMove	16
GecodeSolver	18
GeneralSolver	20
IntegerVariable	22
Invariant	25
Linear	28
LSSpace	30
Model	31
Move	34
Multistop	35
NeighborhoodExplorer	37
Options	38
Random	39
Constraint::SortGreater	40
State	40
Sum	42
ValueChangeMove	44
var	45

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

BP_Data.cpp	47
BP_Data.hpp	47
BPSolver.hpp	48
Clock.cpp	49
Clock.hpp	50
Constants.hpp	50
Constraint.hpp	53
FlipMove.cpp	54
FlipMove.hpp	55
GecodeSolver.cpp	56
GecodeSolver.hpp	57
GeneralSolver.cpp	58
GeneralSolver.hpp	58
getRSS.hpp	60
IntegerVariable.cpp	60
IntegerVariable.hpp	60
Invariant.hpp	61
Linear.hpp	62
LSSpace.cpp	64
LSSpace.hpp	64
main.cpp	65
Model.cpp	66
Model.hpp	67
Move.hpp	68
Multistop.cpp	70
Multistop.hpp	70
NeighborhoodExplorer.cpp	71
NeighborhoodExplorer.hpp	72
Options.cpp	74
Options.hpp	74
Random.cpp	75
Random.hpp	75
State.cpp	76
State.hpp	77
Sum.cpp	78
Sum.hpp	79
ValueChangeMove.cpp	81
ValueChangeMove.hpp	82

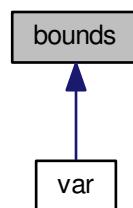
Chapter 4

Class Documentation

4.1 bounds Struct Reference

```
#include <BP_Data.hpp>
```

Inheritance diagram for bounds:



Public Member Functions

- `~bounds()`

Public Attributes

- int `type`
- double `lb`
- double `ub`

4.1.1 Constructor & Destructor Documentation

4.1.1.1 `bounds::~bounds()` `[inline]`

4.1.2 Member Data Documentation

4.1.2.1 `double bounds::lb`

4.1.2.2 int bounds::type

4.1.2.3 double bounds::ub

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

- [BP_Data.hpp](#)

4.2 BP_Input Class Reference

```
#include <BP_Data.hpp>
```

Public Member Functions

- [BP_Input](#) (string file_name)
- [bounds getBterms](#) (const int i) const
- const vector< [elem](#) > & [getMatcoeff](#) (const int i) const
- const vector< [elem](#) > & [getMatcoeff2](#) (const int j) const
- unsigned [getNcons](#) () const
- unsigned [getNvars](#) () const
- var [getVar](#) (const int j) const
- vector< [var](#) > [getVars](#) () const
- int [getDirection](#) () const
- unsigned [getScale](#) () const
- void [setDigits](#) (double number)
- [~BP_Input](#) ()

Protected Attributes

- int [nvars](#)
- int [ncons](#)
- int [nbinvars](#)
- int [nintvars](#)
- int [direction](#)
- unsigned [scale](#)
- vector< vector< [elem](#) > > [matcoeff](#)
- vector< vector< [elem](#) > > [matcoeff2](#)
- vector< [bounds](#) > [bterms](#)
- vector< [var](#) > [vars](#)

Friends

- ostream & [operator<<](#) (ostream &os, const [BP_Input](#) &bs)

4.2.1 Constructor & Destructor Documentation

4.2.1.1 BP_Input::BP_Input (string file_name)

4.2.1.2 BP_Input::~~BP_Input () [inline]

4.2.2 Member Function Documentation

- 4.2.2.1 `bounds BP_Input::getBterms (const int i) const` `[inline]`
- 4.2.2.2 `int BP_Input::getDirection () const` `[inline]`
- 4.2.2.3 `const vector<elem>& BP_Input::getMatcoeff (const int i) const` `[inline]`
- 4.2.2.4 `const vector<elem>& BP_Input::getMatcoeff2 (const int j) const` `[inline]`
- 4.2.2.5 `unsigned BP_Input::getNcons () const` `[inline]`
- 4.2.2.6 `unsigned BP_Input::getNvars () const` `[inline]`
- 4.2.2.7 `unsigned BP_Input::getScale () const` `[inline]`
- 4.2.2.8 `var BP_Input::getVar (const int j) const` `[inline]`
- 4.2.2.9 `vector<var> BP_Input::getVars () const` `[inline]`
- 4.2.2.10 `void BP_Input::setDigits (double number)`

4.2.3 Friends And Related Function Documentation

- 4.2.3.1 `ostream& operator<< (ostream & os, const BP_Input & bs)` `[friend]`

4.2.4 Member Data Documentation

- 4.2.4.1 `vector<bounds> BP_Input::bterms` `[protected]`
- 4.2.4.2 `int BP_Input::direction` `[protected]`
- 4.2.4.3 `vector<vector<elem>> BP_Input::matcoeff` `[protected]`
- 4.2.4.4 `vector<vector<elem>> BP_Input::matcoeff2` `[protected]`
- 4.2.4.5 `int BP_Input::nbinvars` `[protected]`
- 4.2.4.6 `int BP_Input::ncons` `[protected]`
- 4.2.4.7 `int BP_Input::nintvars` `[protected]`
- 4.2.4.8 `int BP_Input::nvars` `[protected]`
- 4.2.4.9 `unsigned BP_Input::scale` `[protected]`
- 4.2.4.10 `vector<var> BP_Input::vars` `[protected]`

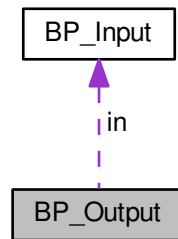
The documentation for this class was generated from the following files:

- [BP_Data.hpp](#)
- [BP_Data.cpp](#)

4.3 BP_Output Class Reference

```
#include <BP_Data.hpp>
```

Collaboration diagram for BP_Output:



Public Member Functions

- [BP_Output](#) (const [BP_Input](#) &i)
- [BP_Output](#) & [operator=](#) (const [BP_Output](#) &out)
- int [assignment](#) (int [var](#)) const
- void [assign](#) (int [var](#), bool b)

Protected Attributes

- const [BP_Input](#) & [in](#)
- vector< bool > [varAssignment](#)

Friends

- ostream & [operator<<](#) (ostream &os, const [BP_Output](#) &out)
- istream & [operator>>](#) (istream &is, [BP_Output](#) &out)

4.3.1 Constructor & Destructor Documentation

4.3.1.1 [BP_Output::BP_Output](#) (const [BP_Input](#) & *i*)

4.3.2 Member Function Documentation

4.3.2.1 void [BP_Output::assign](#) (int *var*, bool *b*)

4.3.2.2 int [BP_Output::assignment](#) (int *var*) const [inline]

4.3.2.3 [BP_Output](#) & [BP_Output::operator=](#) (const [BP_Output](#) & *out*)

4.3.3 Friends And Related Function Documentation

4.3.3.1 ostream& [operator<<](#) (ostream & *os*, const [BP_Output](#) & *out*) [friend]

4.3.3.2 istream& [operator>>](#) (istream & *is*, [BP_Output](#) & *out*) [friend]

4.3.4 Member Data Documentation

4.3.4.1 `const BP_Input& BP_Output::in` `[protected]`

4.3.4.2 `vector<bool> BP_Output::varAssignment` `[protected]`

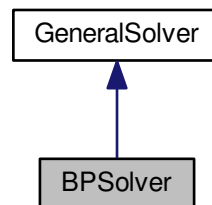
The documentation for this class was generated from the following files:

- [BP_Data.hpp](#)
- [BP_Data.cpp](#)

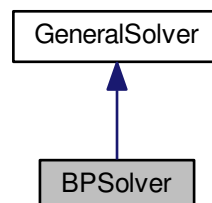
4.4 BPSolver Class Reference

```
#include <BPSolver.hpp>
```

Inheritance diagram for BPSolver:



Collaboration diagram for BPSolver:



Public Member Functions

- [BPSolver](#) ([BP_Input](#) *in)
- [~BPSolver](#) ()
- void [printCurrent](#) ()

Constructor for cloning s.

4.4.1 Constructor & Destructor Documentation

4.4.1.1 `BPSolver::BPSolver (BP_Input * in) [inline]`

4.4.1.2 `BPSolver::~~BPSolver () [inline]`

4.4.2 Member Function Documentation

4.4.2.1 `void BPSolver::printCurrent () [inline]`

Constructor for cloning s.

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

- [BPSolver.hpp](#)

4.5 Clock Class Reference

```
#include <Clock.hpp>
```

Static Public Attributes

- static double [globalClock](#)

4.5.1 Member Data Documentation

4.5.1.1 `double Clock::globalClock [static]`

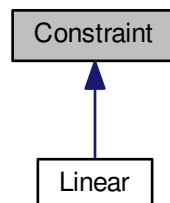
The documentation for this class was generated from the following files:

- [Clock.hpp](#)
- [Clock.cpp](#)

4.6 Constraint Class Reference

```
#include <Constraint.hpp>
```

Inheritance diagram for Constraint:



Classes

- struct [SortGreater](#)

Public Member Functions

- [Constraint](#) ()
- [~Constraint](#) ()
- void [setNumberOfIntegerVariables](#) (int number)
- int & [getNumberOfIntegerVariables](#) ()
- int [getType](#) ()
- int [getDeltaViolation](#) ()
- int [getDeltaViolationDegree](#) ()
- int [getViolation](#) ()
- bool [isOneway](#) ()
- void [isOneway](#) (bool set)
- std::unordered_map< int, [coefType](#) > & [getCoefficients](#) ()
- std::vector< [IntegerVariable](#) * > & [getVariables](#) ()
- int [getViolationDegree](#) ()
- int [getArgument](#) (int i)
- void [setInvariant](#) (std::shared_ptr< [Invariant](#) > invar)
- std::shared_ptr< [Invariant](#) > & [getInvariant](#) ()
- unsigned [getScopeSize](#) ()
- bool [operator<](#) ([Constraint](#) &cons) const
- bool [operator>](#) ([Constraint](#) &cons) const
- virtual int [setDeltaViolation](#) ()
- virtual int [setDeltaViolationDegree](#) ()
- virtual int [updateViolation](#) ()
- virtual int [updateViolationDegree](#) ()
- virtual bool [testCons](#) ()
- virtual bool [testObj](#) ()

Protected Attributes

- int [Violation](#) = 0
- int [ViolationDegree](#) = 0
- int [DeltaViolation](#) = 0
- int [DeltaViolationDegree](#) = 0
- int [priority](#)
- int [type](#)
- bool [oneway](#) = false
- int [numberOfIntegerVariables](#) = 0
- unsigned [scopeSize](#)
- std::vector< int > [arguments](#)
- std::vector< [IntegerVariable](#) * > [variables](#)
- std::unordered_map< int, [coefType](#) > [coefficients](#)
- std::shared_ptr< [Invariant](#) > [invariant](#)

4.6.1 Constructor & Destructor Documentation

4.6.1.1 `Constraint::Constraint ()` [inline]

4.6.1.2 `Constraint::~~Constraint ()` [inline]

4.6.2 Member Function Documentation

4.6.2.1 `int Constraint::getArgument (int i)` [inline]

4.6.2.2 `std::unordered_map<int, coefType>& Constraint::getCoefficients ()` [inline]

4.6.2.3 `int Constraint::getDeltaViolation ()` [inline]

4.6.2.4 `int Constraint::getDeltaViolationDegree ()` [inline]

4.6.2.5 `std::shared_ptr<Invariant>& Constraint::getInvariant ()` [inline]

4.6.2.6 `int& Constraint::getNumberOfIntegerVariables ()` [inline]

4.6.2.7 `unsigned Constraint::getScopeSize ()` [inline]

4.6.2.8 `int Constraint::getType ()` [inline]

4.6.2.9 `std::vector<IntegerVariable*>& Constraint::getVariables ()` [inline]

4.6.2.10 `int Constraint::getViolation ()` [inline]

4.6.2.11 `int Constraint::getViolationDegree ()` [inline]

4.6.2.12 `bool Constraint::isOneway ()` [inline]

4.6.2.13 `void Constraint::isOneway (bool set)` [inline]

4.6.2.14 `bool Constraint::operator< (Constraint & cons) const` [inline]

4.6.2.15 `bool Constraint::operator> (Constraint & cons) const` [inline]

4.6.2.16 `virtual int Constraint::setDeltaViolation ()` [inline], [virtual]

Reimplemented in [Linear](#).

4.6.2.17 `virtual int Constraint::setDeltaViolationDegree ()` [inline], [virtual]

Reimplemented in [Linear](#).

4.6.2.18 `void Constraint::setInvariant (std::shared_ptr< Invariant > invar)` [inline]

4.6.2.19 `void Constraint::setNumberOfIntegerVariables (int number)` [inline]

4.6.2.20 `virtual bool Constraint::testCons ()` [inline], [virtual]

Reimplemented in [Linear](#).

4.6.2.21 `virtual bool Constraint::testObj() [inline],[virtual]`

Reimplemented in [Linear](#).

4.6.2.22 `virtual int Constraint::updateViolation() [inline],[virtual]`

Reimplemented in [Linear](#).

4.6.2.23 `virtual int Constraint::updateViolationDegree() [inline],[virtual]`

Reimplemented in [Linear](#).

4.6.3 Member Data Documentation

4.6.3.1 `std::vector<int> Constraint::arguments [protected]`

4.6.3.2 `std::unordered_map<int,coefType> Constraint::coefficients [protected]`

4.6.3.3 `int Constraint::DeltaViolation = 0 [protected]`

4.6.3.4 `int Constraint::DeltaViolationDegree = 0 [protected]`

4.6.3.5 `std::shared_ptr<Invariant> Constraint::invariant [protected]`

4.6.3.6 `int Constraint::numberOfIntegerVariables = 0 [protected]`

4.6.3.7 `bool Constraint::oneway = false [protected]`

4.6.3.8 `int Constraint::priority [protected]`

4.6.3.9 `unsigned Constraint::scopeSize [protected]`

4.6.3.10 `int Constraint::type [protected]`

4.6.3.11 `std::vector<IntegerVariable*> Constraint::variables [protected]`

4.6.3.12 `int Constraint::Violation = 0 [protected]`

4.6.3.13 `int Constraint::ViolationDegree = 0 [protected]`

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

- [Constraint.hpp](#)

4.7 ConstraintSorter Class Reference

```
#include <Constraint.hpp>
```

Public Member Functions

- [ConstraintSorter](#) ()
- `bool operator()` (std::shared_ptr< [Constraint](#) > &cons1, std::shared_ptr< [Constraint](#) > &cons2)

4.7.1 Constructor & Destructor Documentation

4.7.1.1 `ConstraintSorter::ConstraintSorter ()` `[inline]`

4.7.2 Member Function Documentation

4.7.2.1 `bool ConstraintSorter::operator() (std::shared_ptr< Constraint > & cons1, std::shared_ptr< Constraint > & cons2)` `[inline]`

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

- [Constraint.hpp](#)

4.8 elem Struct Reference

```
#include <BP_Data.hpp>
```

Public Member Functions

- [~elem \(\)](#)

Public Attributes

- int [index](#)
- double [coeff](#)

4.8.1 Constructor & Destructor Documentation

4.8.1.1 `elem::~~elem ()` `[inline]`

4.8.2 Member Data Documentation

4.8.2.1 `double elem::coeff`

4.8.2.2 `int elem::index`

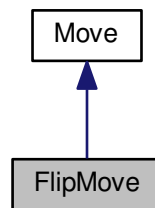
The documentation for this struct was generated from the following file:

- [BP_Data.hpp](#)

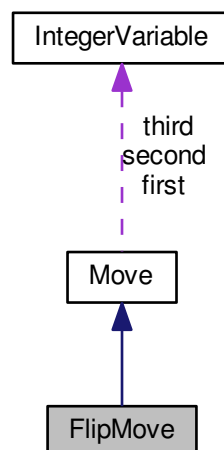
4.9 FlipMove Class Reference

```
#include <FlipMove.hpp>
```


Inheritance diagram for FlipMove:



Collaboration diagram for FlipMove:



Public Member Functions

- `FlipMove` (`IntegerVariable *iv`)
- `FlipMove` (`const FlipMove &orig`)
- `virtual ~FlipMove` ()
- `FlipMove & operator=` (`const FlipMove &a`)

Additional Inherited Members

4.9.1 Constructor & Destructor Documentation

4.9.1.1 `FlipMove::FlipMove (IntegerVariable * iv)`

4.9.1.2 `FlipMove::FlipMove (const FlipMove & orig)`

4.9.1.3 `FlipMove::~~FlipMove ()` `[virtual]`

4.9.2 Member Function Documentation

4.9.2.1 `FlipMove::operator= (const FlipMove & a)`

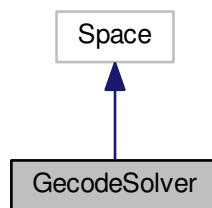
The documentation for this class was generated from the following files:

- [FlipMove.hpp](#)
- [FlipMove.cpp](#)

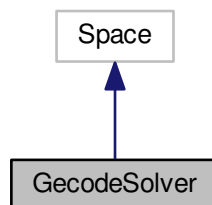
4.10 GecodeSolver Class Reference

```
#include <GecodeSolver.hpp>
```

Inheritance diagram for GecodeSolver:



Collaboration diagram for GecodeSolver:



Public Member Functions

- `GecodeSolver` (`std::shared_ptr< Model > model`)
- `virtual ~GecodeSolver ()`

- void [branch](#) ()
- bool [initialize](#) (int TimeForGecode, bool fix)
- bool [FindSolution](#) (int TimeForGecode, bool fix)
- void [linear](#) (std::vector< int > &coefficients, const std::vector< [IntegerVariable](#) * > &variables, int relation, int upperbound)
- void [createGecodeVariable](#) (int lb, int ub)
- void [SetValues](#) (Gecode::IntVarArray vars)
- void [fixVariables](#) ()
- void [printStatsStatus](#) ()
- void [createArray](#) ()
- void [print_stats](#) (Gecode::Search::Statistics &stat)
- void [print](#) (std::ostream &os) const
- [GecodeSolver](#) (bool share, [GecodeSolver](#) &s)
- void [postCovSol](#) ()
- virtual Space * [copy](#) (bool share)

Protected Attributes

- std::shared_ptr< [Model](#) > [model](#)
- Gecode::IntVarArray [IntVars](#)
- Gecode::IntVarArgs [tmpVars](#)
- Gecode::IntVarArgs [binVars](#)

4.10.1 Constructor & Destructor Documentation

4.10.1.1 [GecodeSolver::GecodeSolver](#) (std::shared_ptr< [Model](#) > *model*)

4.10.1.2 [GecodeSolver::~GecodeSolver](#) () [virtual]

4.10.1.3 [GecodeSolver::GecodeSolver](#) (bool *share*, [GecodeSolver](#) & *s*)

4.10.2 Member Function Documentation

4.10.2.1 void [GecodeSolver::branch](#) ()

4.10.2.2 [Gecode::Space](#) * [GecodeSolver::copy](#) (bool *share*) [virtual]

4.10.2.3 void [GecodeSolver::createArray](#) ()

4.10.2.4 void [GecodeSolver::createGecodeVariable](#) (int *lb*, int *ub*)

4.10.2.5 bool [GecodeSolver::FindSolution](#) (int *TimeForGecode*, bool *fix*)

4.10.2.6 void [GecodeSolver::fixVariables](#) ()

4.10.2.7 bool [GecodeSolver::initialize](#) (int *TimeForGecode*, bool *fix*)

4.10.2.8 void [GecodeSolver::linear](#) (std::vector< int > & *coefficients*, const std::vector< [IntegerVariable](#) * > & *variables*, int *relation*, int *upperbound*)

4.10.2.9 void [GecodeSolver::postCovSol](#) ()

4.10.2.10 void [GecodeSolver::print](#) (std::ostream & *os*) const

4.10.2.11 void GecodeSolver::print_stats (Gecode::Search::Statistics & stat)

4.10.2.12 void GecodeSolver::printSpaceStatus ()

4.10.2.13 void GecodeSolver::SetValues (Gecode::IntVarArray vars)

4.10.3 Member Data Documentation

4.10.3.1 Gecode::IntVarArgs GecodeSolver::binVars [protected]

4.10.3.2 Gecode::IntVarArray GecodeSolver::IntVars [protected]

4.10.3.3 std::shared_ptr<Model> GecodeSolver::model [protected]

4.10.3.4 Gecode::IntVarArgs GecodeSolver::tmpVars [protected]

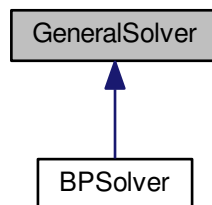
The documentation for this class was generated from the following files:

- [GecodeSolver.hpp](#)
- [GecodeSolver.cpp](#)

4.11 GeneralSolver Class Reference

```
#include <GeneralSolver.hpp>
```

Inheritance diagram for GeneralSolver:



Public Member Functions

- [GeneralSolver](#) ()
- [~GeneralSolver](#) ()
- [GeneralSolver](#) & [operator=](#) (const [GeneralSolver](#) &a)
- void [linear](#) (std::vector< int > &coefficients, std::vector< [IntegerVariable](#) * > &variables, int relation, int ub, unsigned priority)
- void [createIntVars](#) (unsigned numberOfVariables, int lb, int ub)
- void [createIntVar](#) (int lb, int ub)
Create a single variable with given lower and upper bound.
- std::vector< [IntegerVariable](#) * > & [getAllVariables](#) ()
- void [print](#) (std::vector< [IntegerVariable](#) > &IntegerVariables)
Only for testing, should be removed.

- void [InitialSolution](#) (int TimeForGecode)
- void [relax](#) (int timesRelaxed)
- void [simpleRelax](#) (int timesRelaxed)
- void [initializeLS](#) ()
- bool [canBeMadeOneway](#) ([IntegerVariable](#) *iv, [constraint](#) cons)
- void [makeOneway](#) ([IntegerVariable](#) *iv, [constraint](#) cons, int coeff)
- int [getInitialValue](#) ()
- void [optimizeSolution](#) (int time)
- void [printCurrent](#) ()

Private Member Functions

- void [print_stats](#) (Gecode::Search::Statistics &stat)

Private Attributes

- std::shared_ptr< [Model](#) > [model](#) = std::make_shared<[Model](#)> ()
- std::shared_ptr< [State](#) > [st](#) = std::make_shared<[State](#)>(model)
- std::unique_ptr< [LSSpace](#) > [LS](#) = std::unique_ptr<[LSSpace](#)> (new [LSSpace](#)(model))
- std::unique_ptr< [GecodeSolver](#) > [GS](#) = std::unique_ptr<[GecodeSolver](#)> (new [GecodeSolver](#)(model))

Friends

- class [Test](#)

4.11.1 Constructor & Destructor Documentation

4.11.1.1 [GeneralSolver::GeneralSolver](#) () [\[inline\]](#)

4.11.1.2 [GeneralSolver::~~GeneralSolver](#) () [\[inline\]](#)

4.11.2 Member Function Documentation

4.11.2.1 [bool GeneralSolver::canBeMadeOneway](#) ([IntegerVariable](#) * *iv*, [constraint](#) *cons*) [\[inline\]](#)

4.11.2.2 [void GeneralSolver::createIntVar](#) (int *lb*, int *ub*) [\[inline\]](#)

Create a single variable with given lower and upper bound.

4.11.2.3 [void GeneralSolver::createIntVars](#) (unsigned *numberOfVariables*, int *lb*, int *ub*) [\[inline\]](#)

4.11.2.4 [std::vector<IntegerVariable*> &GeneralSolver::getAllVariables](#) () [\[inline\]](#)

4.11.2.5 [int GeneralSolver::getInitialValue](#) () [\[inline\]](#)

4.11.2.6 [void GeneralSolver::initializeLS](#) () [\[inline\]](#)

Sort constraints a variable is part of in decreasing order according to domain

Sort integer variables decreasing order according to number of constraints they are involved

Omskriv det hele. Alt skal bygges op model->invariant fungerer som en prioritet invariant skal opdateres i. Hver variable har deres egen kø.

4.11.2.7 `void GeneralSolver::InitialSolution (int TimeForGecode)` `[inline]`

Uses Gecode to find initial solution returns a new [GeneralSolver](#) with the initial solution the old (the one this method is called from) is not updated with new solution.

4.11.2.8 `void GeneralSolver::linear (std::vector< int > & coefficients, std::vector< IntegerVariable * > & variables, int relation, int ub, unsigned priority)` `[inline]`

4.11.2.9 `void GeneralSolver::makeOneway (IntegerVariable * iv, constraint cons, int coeff)` `[inline]`

4.11.2.10 `GeneralSolver& GeneralSolver::operator= (const GeneralSolver & a)` `[inline]`

4.11.2.11 `void GeneralSolver::optimizeSolution (int time)` `[inline]`

4.11.2.12 `void GeneralSolver::print (std::vector< IntegerVariable > & IntegerVariables)` `[inline]`

Only for testing, should be removed.

4.11.2.13 `void GeneralSolver::print_stats (Gecode::Search::Statistics & stat)` `[inline]`, `[private]`

4.11.2.14 `void GeneralSolver::printCurrent ()` `[inline]`

4.11.2.15 `void GeneralSolver::relax (int timesRelaxed)` `[inline]`

relaxes the space (reduce the number of constraints). Used when Gecode cant find a solution in time. Only works for binary Different relaxation can be chosen (not atm) needs to create a new [GecodeSolver](#) (Space) and recreate some of the calls the user made (those that should not be relaxed).

4.11.2.16 `void GeneralSolver::simpleRelax (int timesRelaxed)` `[inline]`

4.11.3 Friends And Related Function Documentation

4.11.3.1 `friend class Test` `[friend]`

4.11.4 Member Data Documentation

4.11.4.1 `std::unique_ptr<GecodeSolver> GeneralSolver::GS = std::unique_ptr<GecodeSolver> (new GecodeSolver(model))` `[private]`

4.11.4.2 `std::unique_ptr<LSSpace> GeneralSolver::LS = std::unique_ptr<LSSpace> (new LSSpace(model))` `[private]`

4.11.4.3 `std::shared_ptr<Model> GeneralSolver::model = std::make_shared<Model> ()` `[private]`

4.11.4.4 `std::shared_ptr<State> GeneralSolver::st = std::make_shared<State>(model)` `[private]`

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

- [GeneralSolver.hpp](#)

4.12 IntegerVariable Class Reference

```
#include <IntegerVariable.hpp>
```

Public Member Functions

- [VariableInConstraints](#) & [usedInConstraints](#) ()
- void [addToUsedInConstraints](#) (std::shared_ptr< [Constraint](#) > [constraint](#))
- [invariant](#) [getOneway](#) ()
- void [setDefinedBy](#) ([invariant](#) [invar](#), [constraint](#) [cons](#))
- [IntegerVariable](#) (int [lowerbound](#), int [upperbound](#), int [id](#))
- void [setCurrentValue](#) (int [val](#))
- bool [isIntegerVariable](#) ()
- int [getCurrentValue](#) ()
- void [addToUpdate](#) ([updateType](#) [invariant](#))
- [InvariantContainer](#) & [getUpdateVector](#) ()
- int [getID](#) ()
- [Gecode::IntVar](#) * [getVariablePointer](#) ()
- void [setVariablePointer](#) ([Gecode::IntVar](#) &[gecodeVar](#))
- int [getLowerBound](#) ()
- int [getUpperBound](#) ()
- [~IntegerVariable](#) ()

Public Attributes

- int [usedIn](#) = 0
Only used for testing instances.

Protected Member Functions

- void [clearUpdateVector](#) ()

Protected Attributes

- int [lowerBound](#)
- int [upperBound](#)
- int [vectorID](#)
- int [value](#) = 0
- bool [isInteger](#) = false
- bool [isDefined](#) = false
- [VariableInConstraints](#) [constraints](#)
- [invariant](#) [oneway](#)
- [constraint](#) [definedByCons](#)
- [updateVector](#) [update](#)
- [Gecode::IntVarArray](#) * [ArrayPointer](#)
- [Gecode::IntVar](#) * [VariablePointer](#)

Friends

- class [GeneralSolver](#)

4.12.1 Constructor & Destructor Documentation

4.12.1.1 `IntegerVariable::IntegerVariable (int lowerbound, int upperbound, int id)` `[inline]`

4.12.1.2 `IntegerVariable::~~IntegerVariable ()` `[inline]`

4.12.2 Member Function Documentation

4.12.2.1 `void IntegerVariable::addToUpdate (updateType invariant)` `[inline]`

4.12.2.2 `void IntegerVariable::addToUsedInConstraints (std::shared_ptr< Constraint > constraint)` `[inline]`

4.12.2.3 `void IntegerVariable::clearUpdateVector ()` `[inline]`, `[protected]`

4.12.2.4 `int IntegerVariable::getCurrentValue ()` `[inline]`

4.12.2.5 `int IntegerVariable::getID ()` `[inline]`

4.12.2.6 `int IntegerVariable::getLowerBound ()` `[inline]`

4.12.2.7 `invariant IntegerVariable::getOneway ()` `[inline]`

4.12.2.8 `InvariantContainer& IntegerVariable::getUpdateVector ()` `[inline]`

4.12.2.9 `int IntegerVariable::getUpperBound ()` `[inline]`

4.12.2.10 `Gecode::IntVar* IntegerVariable::getVariablePointer ()` `[inline]`

4.12.2.11 `bool IntegerVariable::isIntegerVariable ()` `[inline]`

4.12.2.12 `void IntegerVariable::setCurrentValue (int val)` `[inline]`

4.12.2.13 `void IntegerVariable::setDefinedBy (invariant invar, constraint cons)` `[inline]`

4.12.2.14 `void IntegerVariable::setVariablePointer (Gecode::IntVar & gecodeVar)` `[inline]`

4.12.2.15 `VariableInConstraints& IntegerVariable::usedInConstraints ()` `[inline]`

4.12.3 Friends And Related Function Documentation

4.12.3.1 `friend class GeneralSolver` `[friend]`

4.12.4 Member Data Documentation

4.12.4.1 `Gecode::IntVarArray* IntegerVariable::ArrayPointer` `[protected]`

4.12.4.2 `VariableInConstraints IntegerVariable::constraints` `[protected]`

4.12.4.3 `constraint IntegerVariable::definedByCons` `[protected]`

4.12.4.4 `bool IntegerVariable::isDefined = false` `[protected]`

4.12.4.5 `bool IntegerVariable::isInteger = false` `[protected]`

4.12.4.6 `int IntegerVariable::lowerBound` `[protected]`

4.12.4.7 `invariant IntegerVariable::oneway` `[protected]`

4.12.4.8 `updateVector IntegerVariable::update` `[protected]`

4.12.4.9 `int IntegerVariable::upperBound` `[protected]`

4.12.4.10 `int IntegerVariable::usedIn = 0`

Only used for testing instances.

4.12.4.11 `int IntegerVariable::value = 0` `[protected]`

4.12.4.12 `Gecode::IntVar* IntegerVariable::VariablePointer` `[protected]`

4.12.4.13 `int IntegerVariable::vectorID` `[protected]`

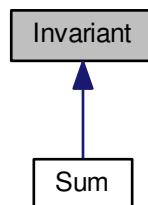
The documentation for this class was generated from the following file:

- [IntegerVariable.hpp](#)

4.13 Invariant Class Reference

```
#include <Invariant.hpp>
```

Inheritance diagram for Invariant:



Public Member Functions

- [Invariant](#) ()
- [Invariant](#) (const [Invariant](#) &a)
- virtual void [addChange](#) (int variableNumber, int change)
- virtual int [calculateDeltaValue](#) ()
- virtual void [initialize](#) ()
- int [getDeltaValue](#) ()
- virtual bool [test](#) ()
- virtual [~Invariant](#) ()
- int [getCurrentValue](#) ()
- void [updateValue](#) ()
- void [setUsedByConstraint](#) (int [constraint](#), int priority)

- void [setUsedByObjective](#) (int objective)
- int [getConstraintNumber](#) ()
- int [getUsedInObjective](#) ()
- unsigned [getPriority](#) ()
- int [getType](#) ()
- [InvariantContainer](#) & [getUpdateVector](#) ()
- void [addToUpdate](#) (invariant invar)
- std::vector< [IntegerVariable](#) * > & [getVariables](#) ()
- std::unordered_map< int, [coefType](#) > [getCoefficients](#) ()
- *Passing by value.*
- int [getVariableID](#) ()
- *Only used when the invariant defines a variable through a oneway constraint.*

Protected Attributes

- std::vector< [IntegerVariable](#) * > [VariablePointers](#)
- *Correspond to the priority the invariant should be updated.*
- double [CurrentValue](#) = 0
- double [DeltaValue](#) = 0
- unsigned [constraintPriority](#)
- int [usedInConstraintNr](#)
- int [usedInObjectiveNr](#)
- std::unordered_map< int, [coefType](#) > [coefficients](#)
- int [type](#)
- int [startValue](#) = 0
- *Should be defined when creating oneway constraints that define (integer)variables.*
- int [variableID](#)
- [InvariantContainer](#) invariants
- [InvariantContainer](#) update

Friends

- class [GeneralSolver](#)

4.13.1 Constructor & Destructor Documentation

- 4.13.1.1 [Invariant::Invariant](#) () [[inline](#)]
- 4.13.1.2 [Invariant::Invariant](#) (const [Invariant](#) & a) [[inline](#)]
- 4.13.1.3 virtual [Invariant::~~Invariant](#) () [[inline](#)],[[virtual](#)]

4.13.2 Member Function Documentation

- 4.13.2.1 virtual void [Invariant::addChange](#) (int *variableNumber*, int *change*) [[inline](#)],[[virtual](#)]

Reimplemented in [Sum](#).

- 4.13.2.2 void [Invariant::addToUpdate](#) (invariant *invar*) [[inline](#)]
- 4.13.2.3 virtual int [Invariant::calculateDeltaValue](#) () [[inline](#)],[[virtual](#)]

Reimplemented in [Sum](#).

4.13.2.4 `std::unordered_map<int, coefType> Invariant::getCoefficients ()` [inline]

Passing by value.

4.13.2.5 `int Invariant::getConstraintNumber ()` [inline]

4.13.2.6 `int Invariant::getCurrentValue ()` [inline]

4.13.2.7 `int Invariant::getDeltaValue ()` [inline]

4.13.2.8 `unsigned Invariant::getPriority ()` [inline]

4.13.2.9 `int Invariant::getType ()` [inline]

4.13.2.10 `InvariantContainer& Invariant::getUpdateVector ()` [inline]

4.13.2.11 `int Invariant::getUsedInObjective ()` [inline]

4.13.2.12 `int Invariant::getVariableID ()` [inline]

Only used when the invariant defines a variable through a oneway constraint.

4.13.2.13 `std::vector<IntegerVariable*>& Invariant::getVariables ()` [inline]

4.13.2.14 `virtual void Invariant::initialize ()` [inline],[virtual]

4.13.2.15 `void Invariant::setUsedByConstraint (int constraint, int priority)` [inline]

4.13.2.16 `void Invariant::setUsedByObjective (int objective)` [inline]

4.13.2.17 `virtual bool Invariant::test ()` [inline],[virtual]

Reimplemented in [Sum](#).

4.13.2.18 `void Invariant::updateValue ()` [inline]

4.13.3 Friends And Related Function Documentation

4.13.3.1 `friend class GeneralSolver` [friend]

4.13.4 Member Data Documentation

4.13.4.1 `std::unordered_map<int, coefType> Invariant::coefficients` [protected]

4.13.4.2 `unsigned Invariant::constraintPriority` [protected]

4.13.4.3 `double Invariant::CurrentValue = 0` [protected]

4.13.4.4 `double Invariant::DeltaValue = 0` [protected]

4.13.4.5 `InvariantContainer Invariant::invariants` [protected]

4.13.4.6 `int Invariant::startValue = 0` `[protected]`

Should be defined when creating oneway constraints that define (integer)variables.

4.13.4.7 `int Invariant::type` `[protected]`

4.13.4.8 `InvariantContainer Invariant::update` `[protected]`

4.13.4.9 `int Invariant::usedInConstraintNr` `[protected]`

4.13.4.10 `int Invariant::usedInObjectiveNr` `[protected]`

4.13.4.11 `int Invariant::variableID` `[protected]`

4.13.4.12 `std::vector<IntegerVariable*> Invariant::VariablePointers` `[protected]`

Correspond to the priority the invariant should be updated.

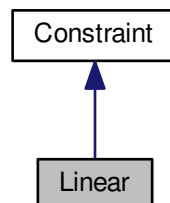
The documentation for this class was generated from the following file:

- [Invariant.hpp](#)

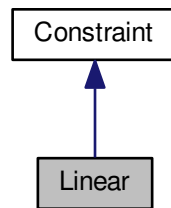
4.14 Linear Class Reference

```
#include <Linear.hpp>
```

Inheritance diagram for Linear:



Collaboration diagram for Linear:



Public Member Functions

- `Linear` (`std::vector< int > &coefficients`, `std::vector< IntegerVariable * > &variables`, `int ub`, `int relation`)
Used to create the original (given by user) constraints.
- `~Linear` ()
Used to create the `Linear` constraint used by local search.
- `int setDeltaViolation` ()
- `int setDeltaViolationDegree` ()
- `int updateViolation` ()
- `int updateViolationDegree` ()
- `bool testCons` ()
- `bool testObj` ()

Protected Attributes

- `int rhs`
- `int relation`

4.14.1 Constructor & Destructor Documentation

4.14.1.1 `Linear::Linear (std::vector< int > & coefficients, std::vector< IntegerVariable * > & variables, int ub, int relation) [inline]`

Used to create the original (given by user) constraints.

4.14.1.2 `Linear::~~Linear () [inline]`

Used to create the `Linear` constraint used by local search.

4.14.2 Member Function Documentation

4.14.2.1 `int Linear::setDeltaViolation () [inline],[virtual]`

Reimplemented from `Constraint`.

4.14.2.2 `int Linear::setDeltaViolationDegree () [inline],[virtual]`

Reimplemented from [Constraint](#).

4.14.2.3 `bool Linear::testCons () [inline],[virtual]`

Reimplemented from [Constraint](#).

4.14.2.4 `bool Linear::testObj () [inline],[virtual]`

Reimplemented from [Constraint](#).

4.14.2.5 `int Linear::updateViolation () [inline],[virtual]`

Reimplemented from [Constraint](#).

4.14.2.6 `int Linear::updateViolationDegree () [inline],[virtual]`

Reimplemented from [Constraint](#).

4.14.3 Member Data Documentation

4.14.3.1 `int Linear::relation [protected]`

4.14.3.2 `int Linear::rhs [protected]`

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

- [Linear.hpp](#)

4.15 LSSpace Class Reference

```
#include <LSSpace.hpp>
```

Public Member Functions

- [LSSpace](#) (std::shared_ptr< [Model](#) > model)
- void [printCurrent](#) ()
- [~LSSpace](#) ()
- void [optimizeSolution](#) (int time, std::shared_ptr< [State](#) > st)
- void [simpleMove](#) (int variabelNr)
- bool [bestImprovement](#) ()

Private Attributes

- int [Violations](#) = 0
- int [iterations](#) = 0
- std::shared_ptr< [Model](#) > model

Friends

- class [Test](#)

4.15.1 Constructor & Destructor Documentation

4.15.1.1 `LSSpace::LSSpace (std::shared_ptr< Model > model)`

4.15.1.2 `LSSpace::~~LSSpace () [inline]`

4.15.2 Member Function Documentation

4.15.2.1 `bool LSSpace::bestImprovement ()`

4.15.2.2 `void LSSpace::optimizeSolution (int time, std::shared_ptr< State > st)`

4.15.2.3 `void LSSpace::printCurrent ()`

4.15.2.4 `void LSSpace::simpleMove (int variableNr)`

4.15.3 Friends And Related Function Documentation

4.15.3.1 `friend class Test [friend]`

4.15.4 Member Data Documentation

4.15.4.1 `int LSSpace::iterations = 0 [private]`

4.15.4.2 `std::shared_ptr<Model> LSSpace::model [private]`

4.15.4.3 `int LSSpace::Violations = 0 [private]`

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

- [LSSpace.hpp](#)
- [LSSpace.cpp](#)

4.16 Model Class Reference

```
#include <Model.hpp>
```

Public Member Functions

- [Model](#) ()
- [Model](#) (const [Model](#) &orig)
- [~Model](#) ()
- void [addBinaryVariable](#) (int lb, int ub)
- void [addIntegerVariable](#) (int lb, int ub)
- [IntegerVariable](#) * [getMaskAt](#) (int i)
- std::vector< [IntegerVariable](#) * > & [getMask](#) ()
- std::list< [IntegerVariable](#) * > & [getIntegerVariables](#) ()
Returns all integer variables as a list. only used to make oneway constraints.
- [variableContainer](#) & [getNonFixedBinaryVariables](#) ()

- [variableContainer](#) & [getAllVariables](#) ()
- [InvariantContainer](#) & [getInvariants](#) ()
- [constraintContainer](#) [getConstraintsWithPriority](#) (int prio)
- [allConstraints](#) & [getConstraints](#) ()
- [constraintContainer](#) [getObjectives](#) ()
- [IntegerVariable](#) * [getNonFixedBinaryVariable](#) (int i)
- void [updateIntegerVariable](#) (int index, Gecode::IntVar &variable)
- void [nonFixedVariables](#) (std::vector< [IntegerVariable](#) * > *nonFixed)

Public Attributes

- int [initialValue](#)
Should be moved to state.

Private Attributes

- [variableContainer](#) original
All variables given by user.
- [variableContainer](#) [nonFixedBinaryVariables](#)
Binary Variables that are not fixed by preprocessing.
- [InvariantContainer](#) [Invariants](#)
Invariants for original model, before oneway constraints are made.
- std::list< [IntegerVariable](#) * > [IntegerVariables](#)
All integer variables in the model.
- [allConstraints](#) [Constraints](#)
- std::vector< [IntegerVariable](#) * > [mask](#)
All Constraints after some has been made oneway (oneway is not included, oneway is invariants)

4.16.1 Constructor & Destructor Documentation

4.16.1.1 [Model::Model](#) ()

4.16.1.2 [Model::Model](#) (const [Model](#) & orig)

4.16.1.3 [Model::~~Model](#) ()

4.16.2 Member Function Documentation

4.16.2.1 void [Model::addBinaryVariable](#) (int lb, int ub)

4.16.2.2 void [Model::addIntegerVariable](#) (int lb, int ub)

4.16.2.3 [variableContainer](#) & [Model::getAllVariables](#) ()

4.16.2.4 [allConstraints](#) & [Model::getConstraints](#) ()

4.16.2.5 [constraintContainer](#) [Model::getConstraintsWithPriority](#) (int prio)

4.16.2.6 std::list< [IntegerVariable](#) * > & [Model::getIntegerVariables](#) ()

Returns all integer variables as a list. only used to make oneway constraints.

4.16.2.7 **InvariantContainer** & Model::getInvariants ()

4.16.2.8 **std::vector< IntegerVariable * >** & Model::getMask ()

4.16.2.9 **IntegerVariable *** Model::getMaskAt (int *i*)

4.16.2.10 **IntegerVariable *** Model::getNonFixedBinaryVariable (int *i*)

4.16.2.11 **variableContainer** & Model::getNonFixedBinaryVariables ()

4.16.2.12 **constraintContainer** Model::getObjectives ()

4.16.2.13 **void** Model::nonFixedVariables (**std::vector< IntegerVariable * >** * *nonFixed*)

4.16.2.14 **void** Model::updateIntegerVariable (int *index*, Gecode::IntVar & *variable*)

4.16.3 Member Data Documentation

4.16.3.1 **allConstraints** Model::Constraints [private]

4.16.3.2 **int** Model::initialValue

Should be moved to state.

4.16.3.3 **std::list<IntegerVariable*>** Model::IntegerVariables [private]

All integer variables in the model.

4.16.3.4 **InvariantContainer** Model::Invariants [private]

Invariants for original model, before oneway constraints are made.

Invariants including oneway constraints but not the invariant from the constraints oneway is made

4.16.3.5 **std::vector<IntegerVariable*>** Model::mask [private]

All Constraints after some has been made oneway (oneway is not included, oneway is invariants)

4.16.3.6 **variableContainer** Model::nonFixedBinaryVariables [private]

Binary Variables that are not fixed by preprocessing.

4.16.3.7 **variableContainer** Model::original [private]

All variables given by user.

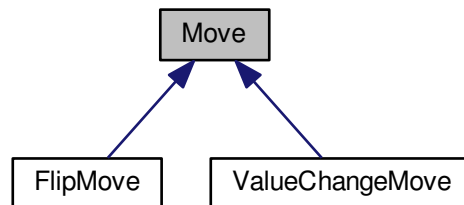
The documentation for this class was generated from the following files:

- [Model.hpp](#)
- [Model.cpp](#)

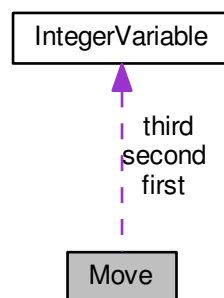
4.17 Move Class Reference

```
#include <Move.hpp>
```

Inheritance diagram for Move:



Collaboration diagram for Move:



Public Member Functions

- `Move ()`
- `Move (IntegerVariable *var1, int delta1, int type)`
- `Move (IntegerVariable *var1, int delta1, IntegerVariable *var2, int delta2, int type)`
- `Move (IntegerVariable *var1, int delta1, IntegerVariable *var2, int delta2, IntegerVariable *var3, int delta3, int type)`
- `void flip ()`
- `~Move ()`
- `Move (const Move &a)`
- `Move & operator= (const Move &a)`
- `void copy (Move *mv)`

Public Attributes

- `IntegerVariable * first`

- int [deltaValueFirst](#)
- [IntegerVariable](#) * [second](#)
- int [deltaValueSecond](#)
- [IntegerVariable](#) * [third](#)
- int [deltaValueThird](#)
- int [moveType](#)

4.17.1 Constructor & Destructor Documentation

4.17.1.1 `Move::Move ()` [\[inline\]](#)

4.17.1.2 `Move::Move (IntegerVariable * var1, int delta1, int type)` [\[inline\]](#)

4.17.1.3 `Move::Move (IntegerVariable * var1, int delta1, IntegerVariable * var2, int delta2, int type)` [\[inline\]](#)

4.17.1.4 `Move::Move (IntegerVariable * var1, int delta1, IntegerVariable * var2, int delta2, IntegerVariable * var3, int delta3, int type)` [\[inline\]](#)

4.17.1.5 `Move::~~Move ()` [\[inline\]](#)

4.17.1.6 `Move::Move (const Move & a)` [\[inline\]](#)

4.17.2 Member Function Documentation

4.17.2.1 `void Move::copy (Move * mv)` [\[inline\]](#)

4.17.2.2 `void Move::flip ()` [\[inline\]](#)

4.17.2.3 `Move& Move::operator= (const Move & a)` [\[inline\]](#)

4.17.3 Member Data Documentation

4.17.3.1 `int Move::deltaValueFirst`

4.17.3.2 `int Move::deltaValueSecond`

4.17.3.3 `int Move::deltaValueThird`

4.17.3.4 `IntegerVariable* Move::first`

4.17.3.5 `int Move::moveType`

4.17.3.6 `IntegerVariable* Move::second`

4.17.3.7 `IntegerVariable* Move::third`

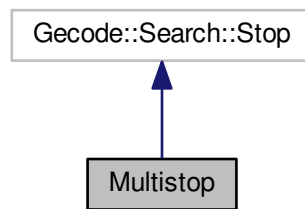
The documentation for this class was generated from the following file:

- [Move.hpp](#)

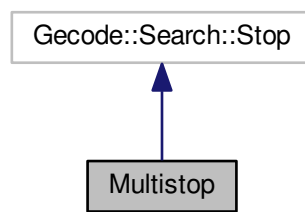
4.18 Multistop Class Reference

```
#include <Multistop.hpp>
```

Inheritance diagram for Multistop:



Collaboration diagram for Multistop:



Public Member Functions

- `Multistop` (unsigned node, unsigned fail, unsigned time)
- virtual bool `stop` (const `Gecode::Search::Statistics` &s, const `Gecode::Search::Options` &o)
Return true if node, time or fail limit is exceeded.
- `Multistop` (const `Multistop` &orig)
- `~Multistop` ()

Public Attributes

- int `called` = 0

Private Attributes

- `Gecode::Search::NodeStop` * `ns`
Used node stop object.
- `Gecode::Search::FailStop` * `fs`
Used fail stop object.
- `Gecode::Search::TimeStop` * `ts`
Used time stop object.

4.18.1 Constructor & Destructor Documentation

4.18.1.1 Multistop::Multistop (unsigned *node*, unsigned *fail*, unsigned *time*)

Creates a Stop object with three stop criteria node, fail, and time. Giving 0 as an argument will ignore that criteria. Time is in ms.

4.18.1.2 Multistop::Multistop (const Multistop & *orig*)

4.18.1.3 Multistop::~Multistop ()

4.18.2 Member Function Documentation

4.18.2.1 bool Multistop::stop (const Gecode::Search::Statistics & *s*, const Gecode::Search::Options & *o*) [virtual]

Return true if node, time or fail limit is exceeded.

4.18.3 Member Data Documentation

4.18.3.1 int Multistop::called = 0

4.18.3.2 Gecode::Search::FailStop* Multistop::fs [private]

Used fail stop object.

4.18.3.3 Gecode::Search::NodeStop* Multistop::ns [private]

Used node stop object.

4.18.3.4 Gecode::Search::TimeStop* Multistop::ts [private]

Used time stop object.

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

- [Multistop.hpp](#)
- [Multistop.cpp](#)

4.19 NeighborhoodExplorer Class Reference

```
#include <NeighborhoodExplorer.hpp>
```

Public Member Functions

- [NeighborhoodExplorer](#) (std::shared_ptr< [Model](#) > *model*)
- [NeighborhoodExplorer](#) (const [NeighborhoodExplorer](#) &*orig*)
- virtual [~NeighborhoodExplorer](#) ()
- void [randomWalk](#) ([Move](#) *mv, std::shared_ptr< [State](#) > st)
- bool [bestImprovement](#) ([Move](#) *mv, std::shared_ptr< [State](#) > st)
- void [makeMove](#) ([Move](#) *mv, std::shared_ptr< [State](#) > st)

Public Attributes

- `std::shared_ptr< Model > model`

Private Member Functions

- `std::vector< int > calculateDeltaChange (Move *mv)`
Not using priority of constraints yet.
- `void commitMove (Move *mv, std::shared_ptr< State > st)`

4.19.1 Constructor & Destructor Documentation

- 4.19.1.1 `NeighborhoodExplorer::NeighborhoodExplorer (std::shared_ptr< Model > model)`
- 4.19.1.2 `NeighborhoodExplorer::NeighborhoodExplorer (const NeighborhoodExplorer & orig)`
- 4.19.1.3 `NeighborhoodExplorer::~~NeighborhoodExplorer ()` [virtual]

4.19.2 Member Function Documentation

- 4.19.2.1 `bool NeighborhoodExplorer::bestImprovement (Move * mv, std::shared_ptr< State > st)`
- 4.19.2.2 `std::vector< int > NeighborhoodExplorer::calculateDeltaChange (Move * mv)` [private]

Not using priority of constraints yet.

- 4.19.2.3 `void NeighborhoodExplorer::commitMove (Move * mv, std::shared_ptr< State > st)` [private]
- 4.19.2.4 `void NeighborhoodExplorer::makeMove (Move * mv, std::shared_ptr< State > st)`
- 4.19.2.5 `void NeighborhoodExplorer::randomWalk (Move * mv, std::shared_ptr< State > st)`

4.19.3 Member Data Documentation

- 4.19.3.1 `std::shared_ptr<Model> NeighborhoodExplorer::model`

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

- [NeighborhoodExplorer.hpp](#)
- [NeighborhoodExplorer.cpp](#)

4.20 Options Class Reference

```
#include <Options.hpp>
```

Public Member Functions

- [Options](#) ()
- [Options](#) (const [Options](#) &orig)
- virtual [~Options](#) ()

Private Attributes

- `int timelimit = 0`

4.20.1 Constructor & Destructor Documentation

4.20.1.1 `Options::Options ()`

4.20.1.2 `Options::Options (const Options & orig)`

4.20.1.3 `Options::~~Options ()` `[virtual]`

4.20.2 Member Data Documentation

4.20.2.1 `int Options::timelimit = 0` `[private]`

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

- [Options.hpp](#)
- [Options.cpp](#)

4.21 Random Class Reference

```
#include <Random.hpp>
```

Static Public Member Functions

- static `int Integer (int lb, int ub)`
- static `int Integer (int ub)`
- static `double Double (double lb=0, double ub=1)`
- static `int Seed (int seed)`

Static Public Attributes

- static `std::mt19937 mt`
- static `int seed = std::random_device()()`

4.21.1 Member Function Documentation

4.21.1.1 `double Random::Double (double lb = 0, double ub = 1)` `[static]`

4.21.1.2 `int Random::Integer (int lb, int ub)` `[static]`

4.21.1.3 `int Random::Integer (int ub)` `[static]`

4.21.1.4 `int Random::Seed (int seed)` `[static]`

4.21.2 Member Data Documentation

4.21.2.1 `std::mt19937 Random::mt` `[static]`

4.21.2.2 `int Random::seed = std::random_device>()` `[static]`

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

- [Random.hpp](#)
- [Random.cpp](#)

4.22 Constraint::SortGreater Struct Reference

```
#include <Constraint.hpp>
```

Public Member Functions

- `bool operator()` (`const std::shared_ptr< Constraint > &cons1`, `const std::shared_ptr< Constraint > &cons2`) `const`

4.22.1 Member Function Documentation

4.22.1.1 `bool Constraint::SortGreater::operator()` (`const std::shared_ptr< Constraint > &cons1`, `const std::shared_ptr< Constraint > &cons2`) `const` `[inline]`

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

- [Constraint.hpp](#)

4.23 State Class Reference

```
#include <State.hpp>
```

Public Member Functions

- `State` (`std::shared_ptr< Model > model`)
- `State` (`const State &orig`)
- `virtual ~State` ()
- `void initializeInvariants` ()
Maybe all the initialize should be moved to model (again).
- `void initializeConstraints` ()
- `void initializeObjective` ()
- `int getObjectiveValue` ()
- `void saveSolution` ()
- `std::vector< int > * getSolutions` ()
- `int getSolutionsValue` ()
- `void setSolution` ()
- `bool recalculateAll` ()
- `int maskAt` (int i)
Move this to model.
- `void shuffleMask` ()
Move this to model.

Public Attributes

- int [numberOfViolations](#)

Private Attributes

- int [solutionValue](#)
- std::vector< int > * [mask](#)
- std::shared_ptr< [Model](#) > [model](#)
- std::vector< int > * [solution](#)

4.23.1 Constructor & Destructor Documentation

4.23.1.1 `State::State (std::shared_ptr< Model > model)`

4.23.1.2 `State::State (const State & orig)`

4.23.1.3 `State::~~State ()` `[virtual]`

4.23.2 Member Function Documentation

4.23.2.1 `int State::getObjectiveValue ()`

4.23.2.2 `std::vector< int > * State::getSolution ()`

4.23.2.3 `int State::getSolutionValue ()`

4.23.2.4 `void State::initializeConstraints ()`

4.23.2.5 `void State::initializeInvariants ()`

Maybe all the initialize should be moved to model (again).

4.23.2.6 `void State::initializeObjective ()`

4.23.2.7 `int State::maskAt (int i)`

[Move](#) this to model.

4.23.2.8 `bool State::recalculateAll ()`

4.23.2.9 `void State::saveSolution ()`

4.23.2.10 `void State::setSolution ()`

4.23.2.11 `void State::shuffleMask ()`

[Move](#) this to model.

4.23.3 Member Data Documentation

4.23.3.1 `std::vector<int>* State::mask` `[private]`

4.23.3.2 `std::shared_ptr<Model> State::model` [private]

4.23.3.3 `int State::numberOfViolations`

4.23.3.4 `std::vector<int>* State::solution` [private]

4.23.3.5 `int State::solutionValue` [private]

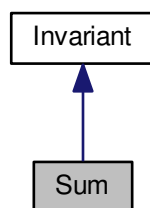
The documentation for this class was generated from the following files:

- [State.hpp](#)
- [State.cpp](#)

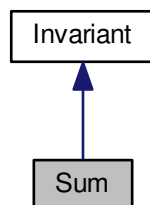
4.24 Sum Class Reference

```
#include <Sum.hpp>
```

Inheritance diagram for Sum:



Collaboration diagram for Sum:



Public Member Functions

- `Sum` (`std::vector< IntegerVariable * > &vars`, `std::vector< int > &c`)
- `Sum` (`std::vector< IntegerVariable * > &vars`, `std::unordered_map< int, coefType > &map`)

Construct that copies a Coefficient map. Size can be different when this sum contains invariants.

- [Sum](#) (const [Sum](#) &a)
- [~Sum](#) ()
- [Sum & operator=](#) (const [Sum](#) &a)
- int [calculateDeltaValue](#) ()
- void [addChange](#) (int variableNumber, int changeInValue)
- bool [test](#) ()

*update currentValue by adding currentValue*coeff of all variables and invariants*

Protected Attributes

- std::vector< int > [VariableChange](#)

4.24.1 Constructor & Destructor Documentation

4.24.1.1 [Sum::Sum](#) (std::vector< [IntegerVariable](#) * > & vars, std::vector< int > & c)

4.24.1.2 [Sum::Sum](#) (std::vector< [IntegerVariable](#) * > & vars, std::unordered_map< int, [coefType](#) > & map)

Construct that copies a Coefficient map. Size can be different when this sum contains invariants.

4.24.1.3 [Sum::Sum](#) (const [Sum](#) & a)

4.24.1.4 [Sum::~~Sum](#) ()

4.24.2 Member Function Documentation

4.24.2.1 void [Sum::addChange](#) (int variableNumber, int changeInValue) [\[virtual\]](#)

Reimplemented from [Invariant](#).

4.24.2.2 int [Sum::calculateDeltaValue](#) () [\[virtual\]](#)

Reimplemented from [Invariant](#).

4.24.2.3 [Sum& Sum::operator=](#) (const [Sum](#) & a)

4.24.2.4 bool [Sum::test](#) () [\[virtual\]](#)

update currentValue by adding currentValue*coeff of all variables and invariants

Reimplemented from [Invariant](#).

4.24.3 Member Data Documentation

4.24.3.1 std::vector<int> [Sum::VariableChange](#) [\[protected\]](#)

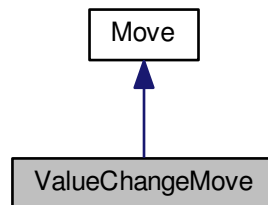
The documentation for this class was generated from the following files:

- [Sum.hpp](#)
- [Sum.cpp](#)

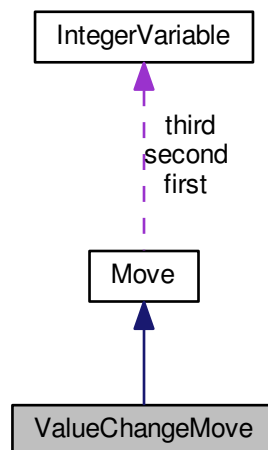
4.25 ValueChangeMove Class Reference

```
#include <ValueChangeMove.hpp>
```

Inheritance diagram for ValueChangeMove:



Collaboration diagram for ValueChangeMove:



Public Member Functions

- [ValueChangeMove](#) ([IntegerVariable](#) *iv, int newValue)
- [ValueChangeMove](#) (const [ValueChangeMove](#) &orig)
- [~ValueChangeMove](#) ()
- [ValueChangeMove](#) & [operator=](#) (const [ValueChangeMove](#) &a)

Additional Inherited Members

4.25.1 Constructor & Destructor Documentation

4.25.1.1 `ValueChangeMove::ValueChangeMove (IntegerVariable * iv, int newValue)`

4.25.1.2 `ValueChangeMove::ValueChangeMove (const ValueChangeMove & orig)`

4.25.1.3 `ValueChangeMove::~~ValueChangeMove ()`

4.25.2 Member Function Documentation

4.25.2.1 `ValueChangeMove & ValueChangeMove::operator= (const ValueChangeMove & a)`

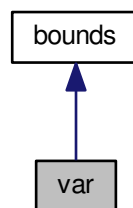
The documentation for this class was generated from the following files:

- [ValueChangeMove.hpp](#)
- [ValueChangeMove.cpp](#)

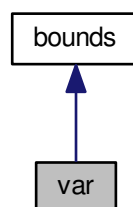
4.26 var Struct Reference

```
#include <BP_Data.hpp>
```

Inheritance diagram for var:



Collaboration diagram for var:



Public Member Functions

- [~var\(\)](#)

Public Attributes

- double [objcoeff](#)
- int [kind](#)
- bool [bin](#)

4.26.1 Constructor & Destructor Documentation

4.26.1.1 `var::~~var ()` [[inline](#)]

4.26.2 Member Data Documentation

4.26.2.1 `bool var::bin`

4.26.2.2 `int var::kind`

4.26.2.3 `double var::objcoeff`

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

- [BP_Data.hpp](#)

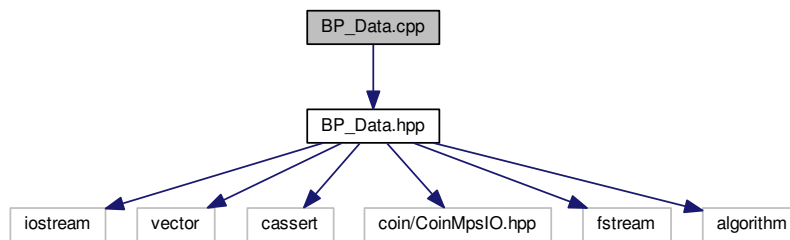
Chapter 5

File Documentation

5.1 BP_Data.cpp File Reference

```
#include "BP_Data.hpp"
```

Include dependency graph for BP_Data.cpp:



Functions

- ostream & [operator<<](#) (ostream &os, const [BP_Output](#) &out)
- istream & [operator>>](#) (istream &is, [BP_Output](#) &out)

5.1.1 Function Documentation

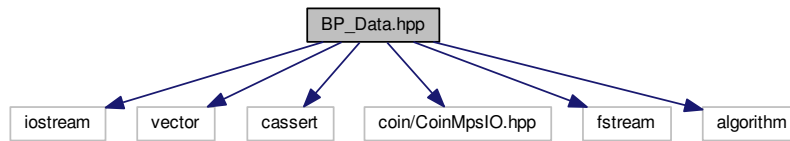
5.1.1.1 ostream& [operator<<](#) (ostream &os, const [BP_Output](#) &out)

5.1.1.2 istream& [operator>>](#) (istream &is, [BP_Output](#) &out)

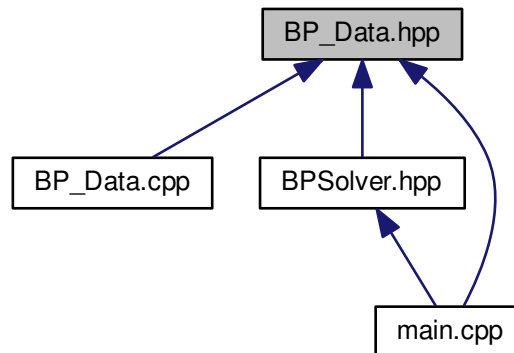
5.2 BP_Data.hpp File Reference

```
#include <iostream>
#include <vector>
#include <cassert>
#include "coin/CoinMpsIO.hpp"
#include <fstream>
#include <algorithm>
```

Include dependency graph for BP_Data.hpp:



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



Classes

- struct [elem](#)
- struct [bounds](#)
- struct [var](#)
- class [BP_Input](#)
- class [BP_Output](#)

Macros

- `#define DEBUG\(X\) X`

5.2.1 Macro Definition Documentation

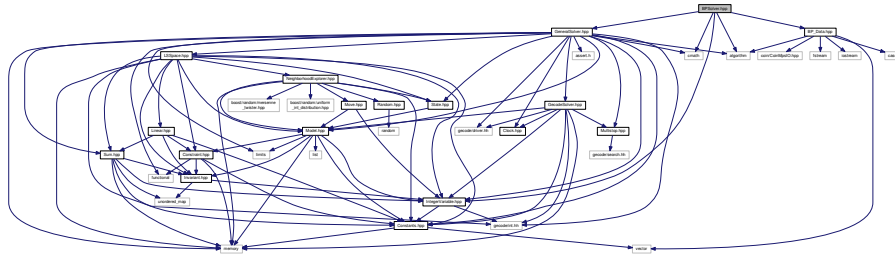
5.2.1.1 `#define DEBUG\(X \) X`

5.3 BPSolver.hpp File Reference

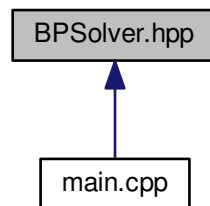
```
#include "BP_Data.hpp"
```



```
#include <cmath>
#include <algorithm>
#include "GeneralSolver.hpp"
#include "IntegerVariable.hpp"
Include dependency graph for BPSolver.hpp:
```



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



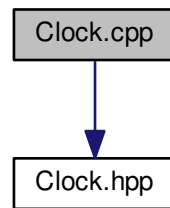
Classes

- class [BPSolver](#)

5.4 Clock.cpp File Reference

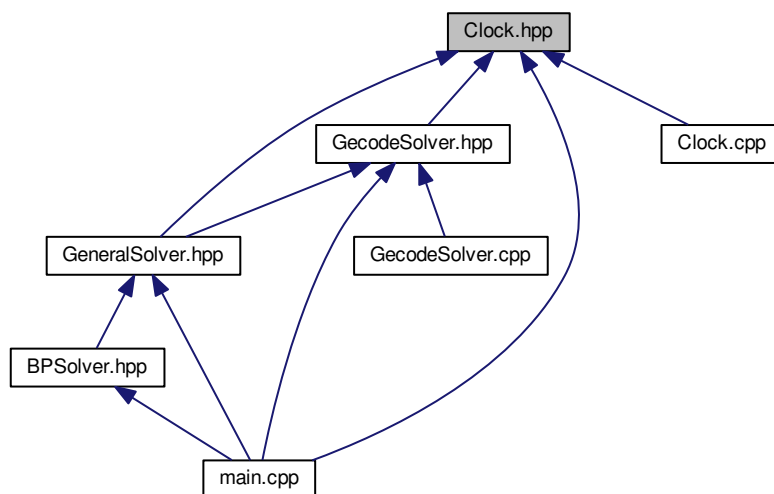
```
#include "Clock.hpp"
```

Include dependency graph for Clock.cpp:



5.5 Clock.hpp File Reference

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



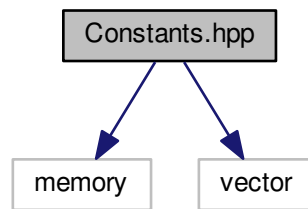
Classes

- class [Clock](#)

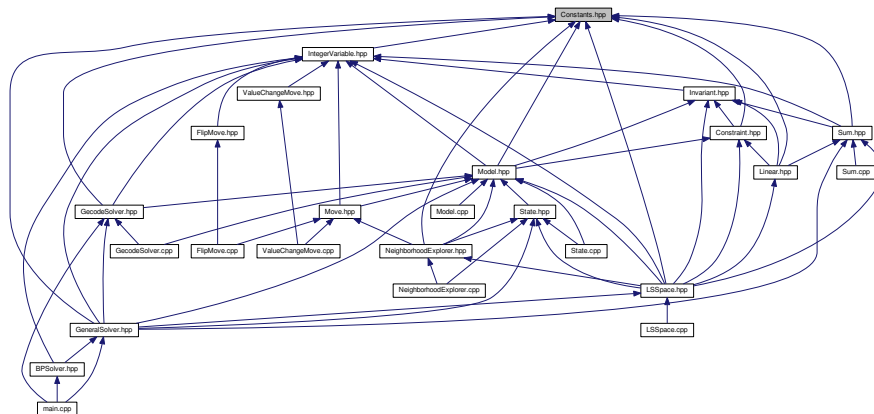
5.6 Constants.hpp File Reference

```
#include <memory>
#include <vector>
```

Include dependency graph for Constants.hpp:



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



Macros

- #define `nonTerm` 0.000000000001
- #define `weight` 100
- #define `tabulistLB` 5
- #define `tabulistUB` 20
- #define `maxIter` 200000000
- #define `iniTrails` 20
- #define `maxMoves` 600
- #define `RANDOMSEED` 1337
- #define `PRINT` TRUE
- #define `debug` `std::cout << __FILE__ << " " << __FUNCTION__ << " " << __LINE__ << std::endl;`
- #define `EQ` 0
- #define `LQ` 1
- #define `GQ` 3
- #define `GR` 4
- #define `LE` 2
- #define `OBJ` 0
- #define `HARD` 1
- #define `SOFT` 2

- `#define FLIP 1`
- `#define SWAP 2`
- `#define VALUECHANGE 3`
- `#define LINEAR 1`
- `#define SUM 1`

Typedefs

- `typedef double coefType`
- `typedef std::shared_ptr
 < Constraint > constraint`
- `typedef std::vector
 < IntegerVariable * > variableContainer`
- `typedef std::vector< constraint > VariableInConstraints`
- `typedef std::shared_ptr
 < Invariant > invariant`
- `typedef std::vector< invariant > InvariantContainer`
- `typedef InvariantContainer updateVector`
- `typedef std::shared_ptr
 < Invariant > updateType`
- `typedef std::shared_ptr
 < std::vector< constraint > > constraintContainer`
- `typedef std::vector
 < constraintContainer > allConstraints`

5.6.1 Macro Definition Documentation

5.6.1.1 `#define debug std::cout << __FILE__ << " " << __FUNCTION__ << " " << __LINE__ << std::endl;`

5.6.1.2 `#define EQ 0`

5.6.1.3 `#define FLIP 1`

5.6.1.4 `#define GQ 3`

5.6.1.5 `#define GR 4`

5.6.1.6 `#define HARD 1`

5.6.1.7 `#define iniTrails 20`

5.6.1.8 `#define LE 2`

5.6.1.9 `#define LINEAR 1`

5.6.1.10 `#define LQ 1`

5.6.1.11 `#define maxIter 200000000`

5.6.1.12 `#define maxMoves 600`

5.6.1.13 `#define nonTerm 0.0000000000001`

5.6.1.14 `#define OBJ 0`

5.6.1.15 `#define PRINT TRUE`

5.6.1.16 `#define RANDOMSEED 1337`

5.6.1.17 `#define SOFT 2`

5.6.1.18 `#define SUM 1`

5.6.1.19 `#define SWAP 2`

5.6.1.20 `#define tabulistLB 5`

5.6.1.21 `#define tabulistUB 20`

5.6.1.22 `#define VALUECHANGE 3`

5.6.1.23 `#define weight 100`

5.6.2 Typedef Documentation

5.6.2.1 `typedef std::vector<constraintContainer> allConstraints`

5.6.2.2 `typedef double coefType`

5.6.2.3 `typedef std::shared_ptr<Constraint> constraint`

5.6.2.4 `typedef std::shared_ptr<std::vector<constraint> > constraintContainer`

5.6.2.5 `typedef std::shared_ptr<Invariant> invariant`

5.6.2.6 `typedef std::vector<invariant> InvariantContainer`

5.6.2.7 `typedef std::shared_ptr<Invariant> updateType`

5.6.2.8 `typedef InvariantContainer updateVector`

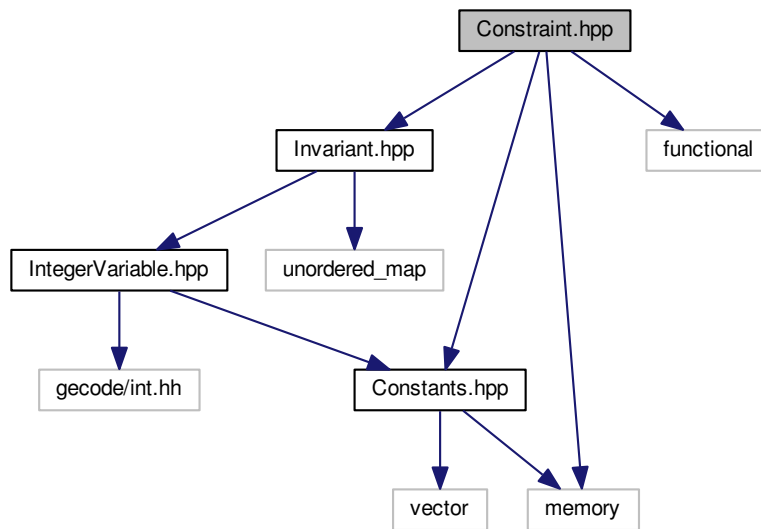
5.6.2.9 `typedef std::vector<IntegerVariable*> variableContainer`

5.6.2.10 `typedef std::vector<constraint> VariableInConstraints`

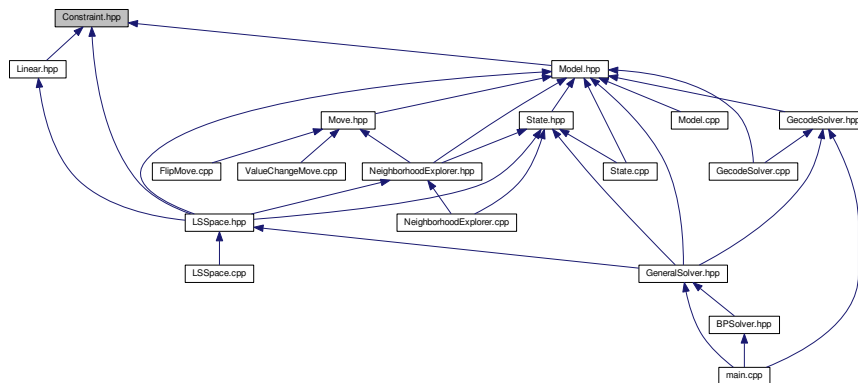
5.7 Constraint.hpp File Reference

```
#include "Invariant.hpp"
#include <memory>
#include <functional>
#include "Constants.hpp"
```

Include dependency graph for Constraint.hpp:



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



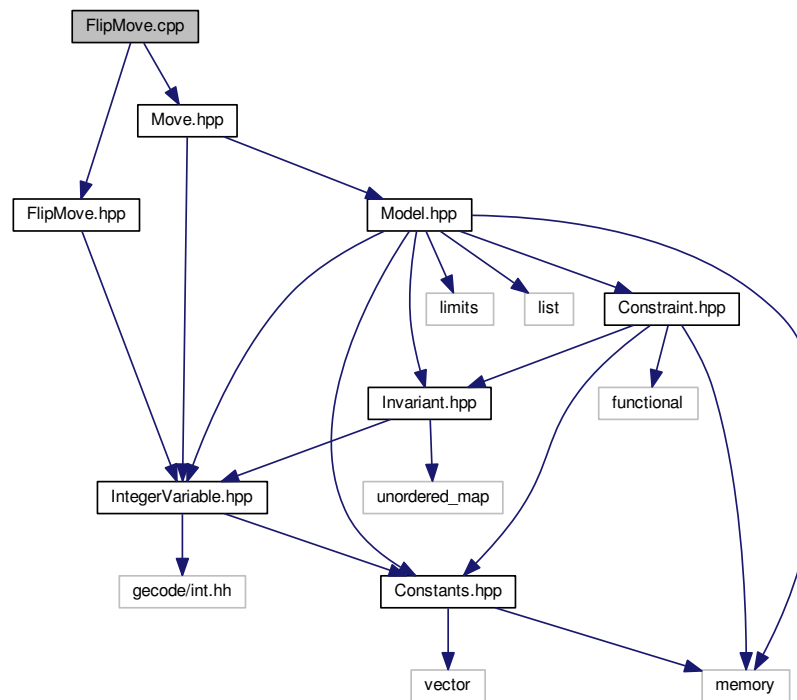
Classes

- class [Constraint](#)
- struct [Constraint::SortGreater](#)
- class [ConstraintSorter](#)

5.8 FlipMove.cpp File Reference

```
#include "FlipMove.hpp"
#include "Move.hpp"
```

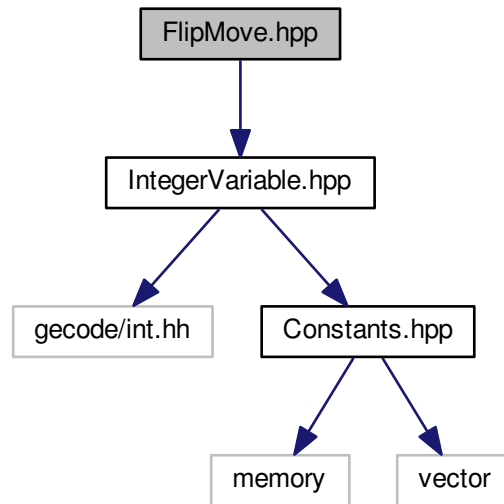
Include dependency graph for FlipMove.cpp:



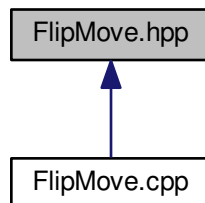
5.9 FlipMove.hpp File Reference

```
#include "IntegerVariable.hpp"
```

Include dependency graph for FlipMove.hpp:



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



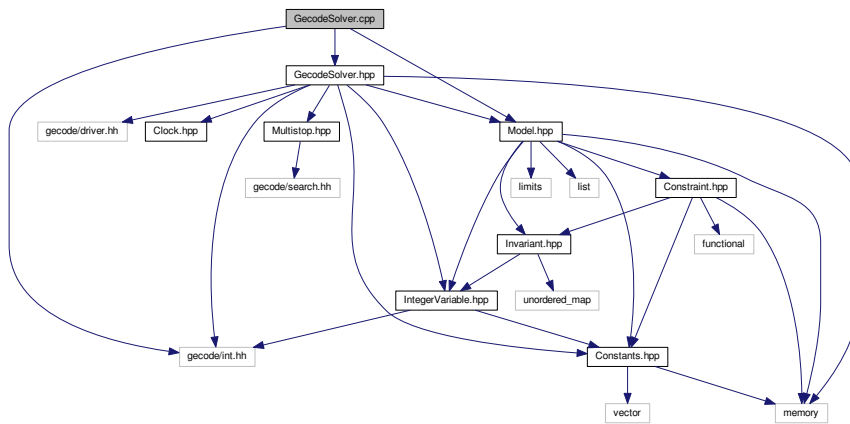
Classes

- class [FlipMove](#)

5.10 GecodeSolver.cpp File Reference

```
#include <gecode/int.hh>
#include "GecodeSolver.hpp"
#include "Model.hpp"
```


Include dependency graph for GecodeSolver.cpp:



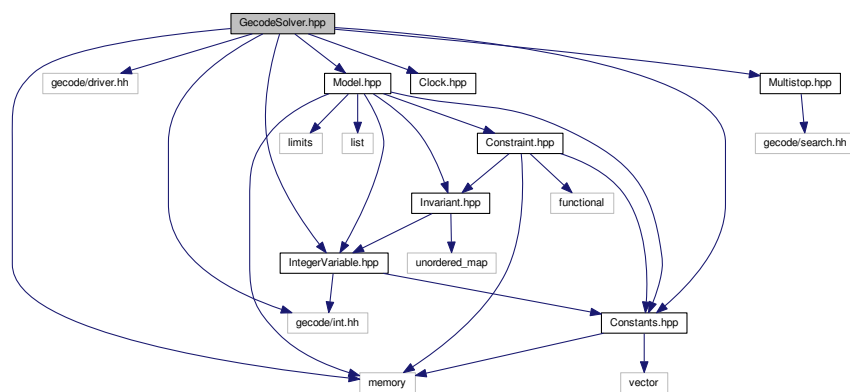
5.11 GecodeSolver.hpp File Reference

```

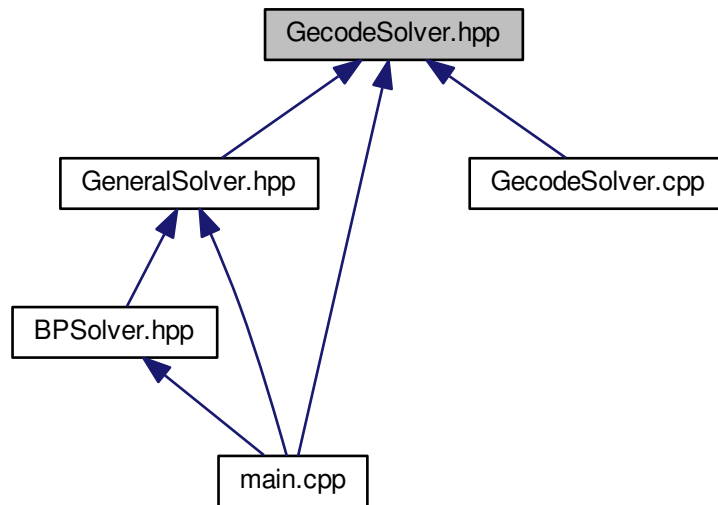
#include <memory>
#include <gecode/driver.hh>
#include <gecode/int.hh>
#include "IntegerVariable.hpp"
#include "Clock.hpp"
#include "Constants.hpp"
#include "Model.hpp"
#include "Multistop.hpp"

```

Include dependency graph for GecodeSolver.hpp:



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



Classes

- class [GecodeSolver](#)

5.12 GeneralSolver.cpp File Reference

5.13 GeneralSolver.hpp File Reference

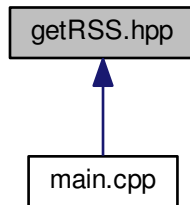
```

#include <cmath>
#include <algorithm>
#include "LSSpace.hpp"
#include <assert.h>
#include "Model.hpp"
#include <gecode/driver.hh>
#include <gecode/int.hh>
#include "Constants.hpp"
#include "GecodeSolver.hpp"
#include <limits>
#include "IntegerVariable.hpp"
#include "Multistop.hpp"
#include "Sum.hpp"
#include <memory>
#include "State.hpp"
#include <functional>
#include "Clock.hpp"

```


5.14 getRSS.hpp File Reference

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



Functions

- `size_t` [getPeakRSS\(\)](#)
- `size_t` [getCurrentRSS\(\)](#)

5.14.1 Function Documentation

5.14.1.1 `size_t` [getCurrentRSS\(\)](#)

Returns the current resident set size (physical memory use) measured in bytes, or zero if the value cannot be determined on this OS.

5.14.1.2 `size_t` [getPeakRSS\(\)](#)

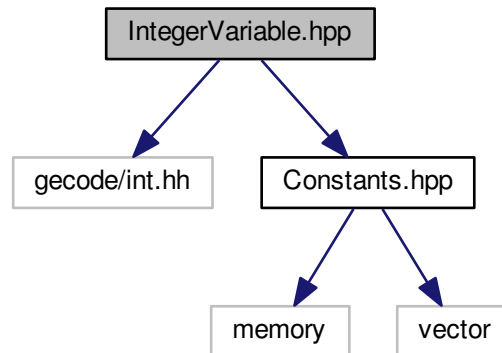
Returns the peak (maximum so far) resident set size (physical memory use) measured in bytes, or zero if the value cannot be determined on this OS.

5.15 IntegerVariable.cpp File Reference

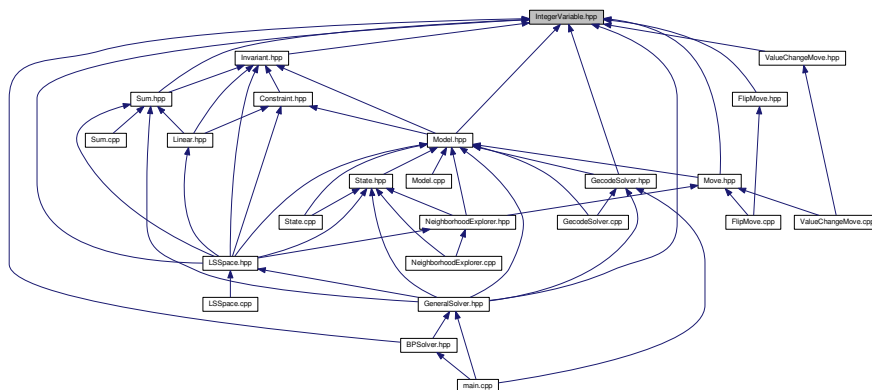
5.16 IntegerVariable.hpp File Reference

```
#include <gecode/int.hh>
#include "Constants.hpp"
```

Include dependency graph for IntegerVariable.hpp:



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



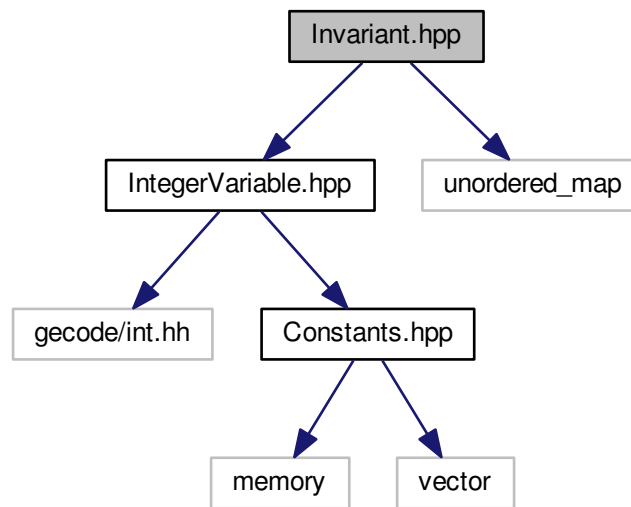
Classes

- class [IntegerVariable](#)

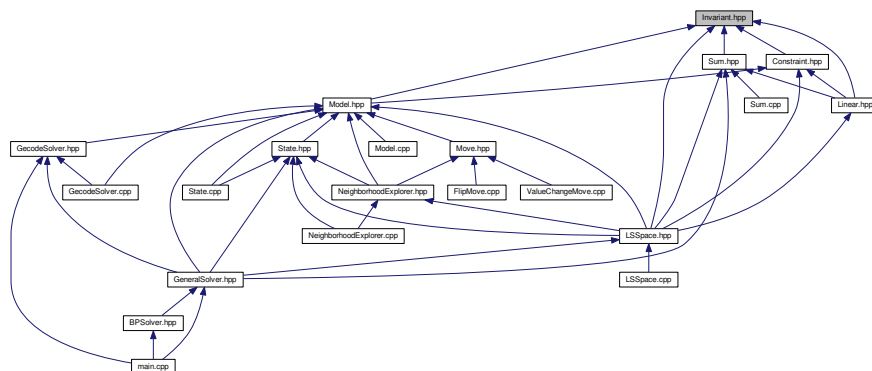
5.17 Invariant.hpp File Reference

```
#include "IntegerVariable.hpp"
#include <unordered_map>
```

Include dependency graph for Invariant.hpp:



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



Classes

- class [Invariant](#)

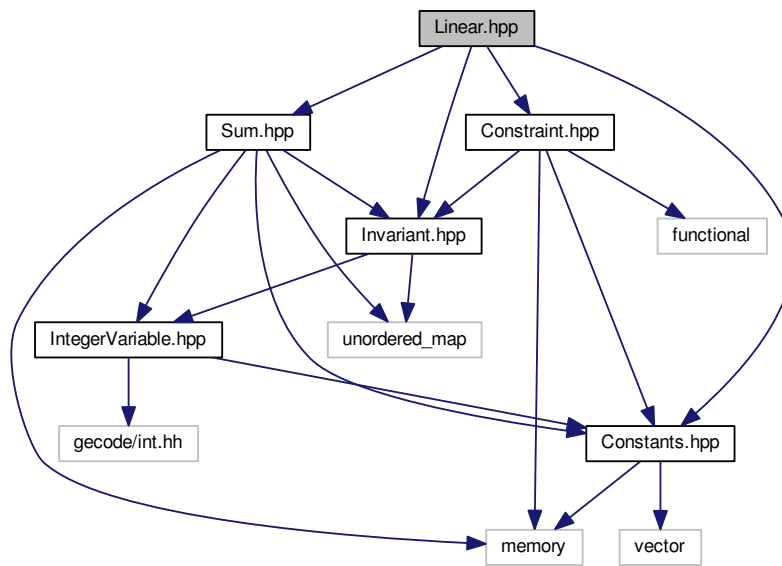
5.18 Linear.hpp File Reference

```

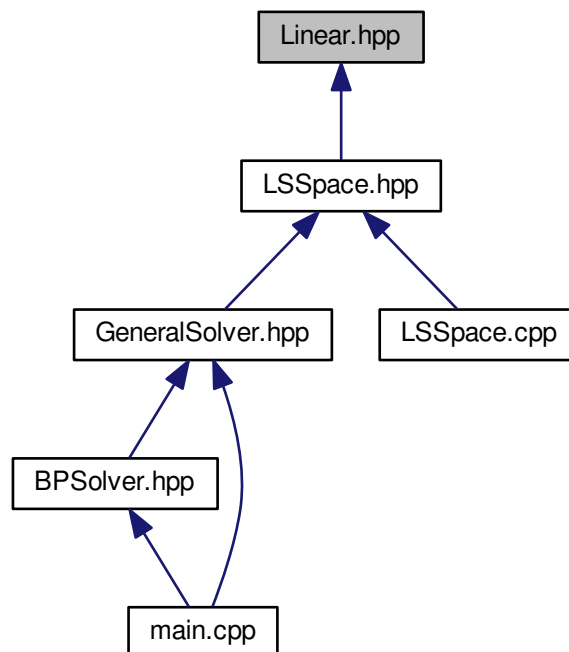
#include "Sum.hpp"
#include "Invariant.hpp"
#include "Constraint.hpp"
#include "Constants.hpp"

```

Include dependency graph for Linear.hpp:



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



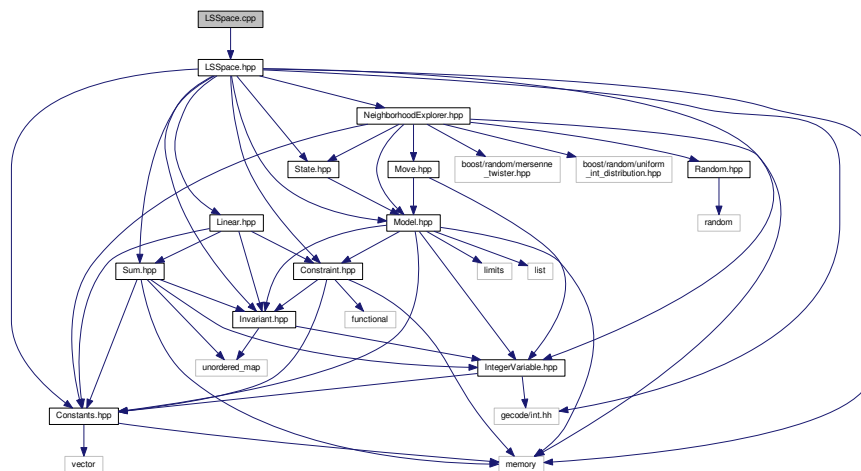
Classes

- class [Linear](#)

5.19 LSSpace.cpp File Reference

```
#include "LSSpace.hpp"
```

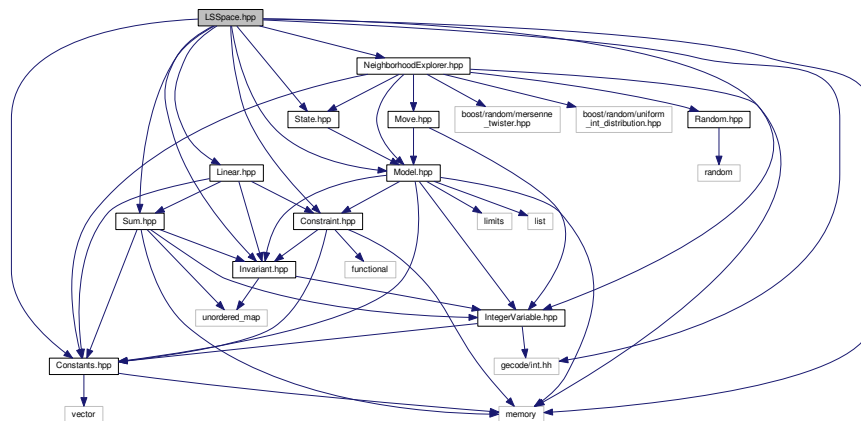
Include dependency graph for LSSpace.cpp:



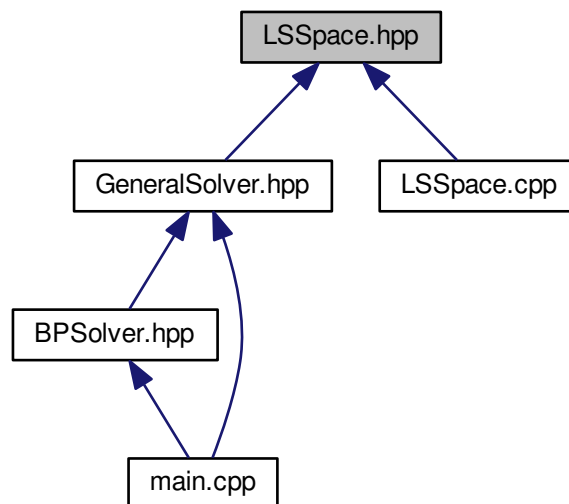
5.20 LSSpace.hpp File Reference

```
#include <geocode/int.hh>
#include "IntegerVariable.hpp"
#include "Invariant.hpp"
#include "Constraint.hpp"
#include "Sum.hpp"
#include "Linear.hpp"
#include "Model.hpp"
#include "NeighborhoodExplorer.hpp"
#include "Constants.hpp"
#include <memory>
#include "State.hpp"
```


Include dependency graph for LSSpace.hpp:



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



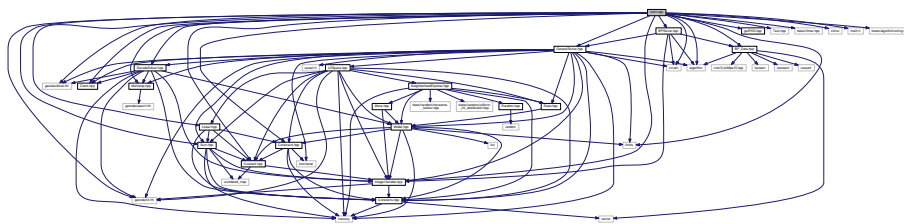
Classes

- class [LSSpace](#)

5.21 main.cpp File Reference

```
#include <gecode/driver.hh>
```

```
#include <gcode/int.hh>
#include "BP_Data.hpp"
#include <cmath>
#include <algorithm>
#include <limits>
#include "BPSolver.hpp"
#include "Clock.hpp"
#include "getRSS.hpp"
#include "GeneralSolver.hpp"
#include "GecodeSolver.hpp"
#include "Test.hpp"
#include "Multistop.hpp"
#include <boost/algorithm/string.hpp>
Include dependency graph for main.cpp:
```



Functions

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

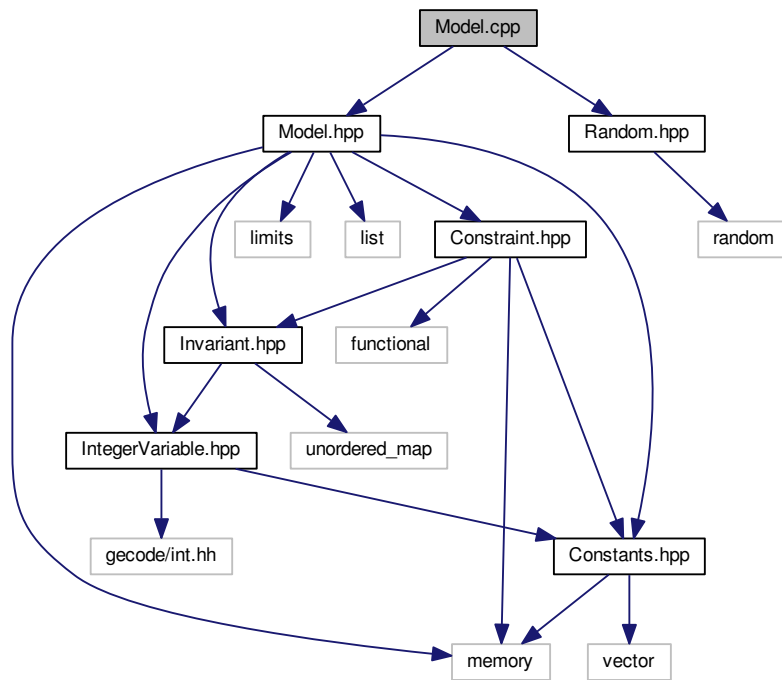
5.21.1 Function Documentation

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

5.22 Model.cpp File Reference

```
#include "Model.hpp"
#include "Random.hpp"
```

Include dependency graph for Model.cpp:



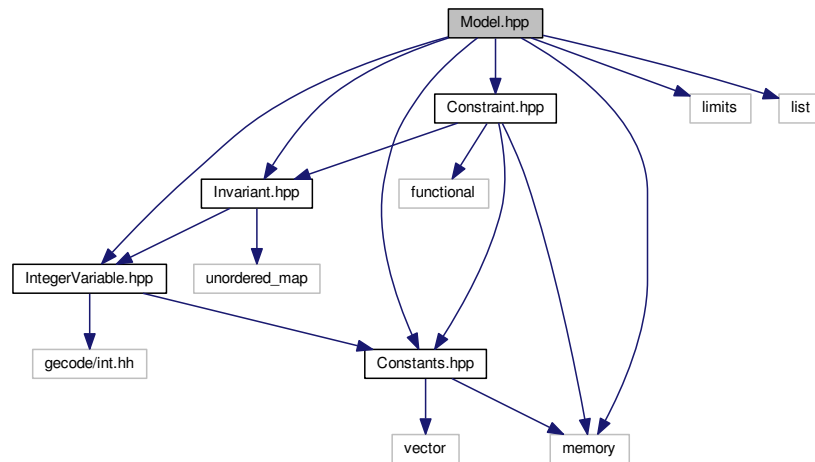
5.23 Model.hpp File Reference

```

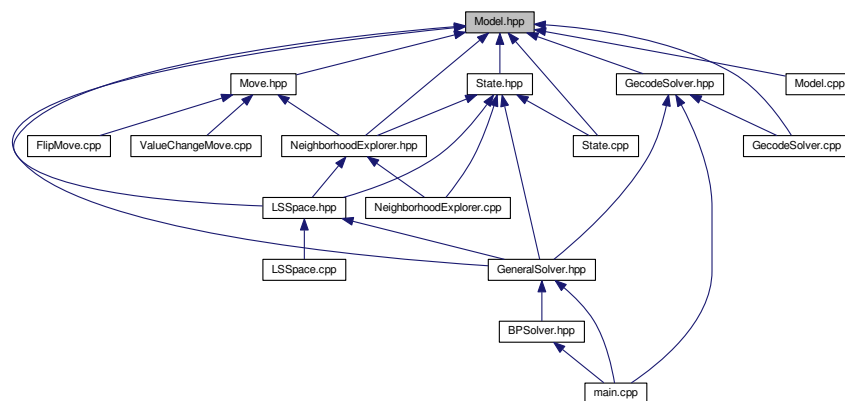
#include "IntegerVariable.hpp"
#include "Invariant.hpp"
#include "Constraint.hpp"
#include "Constants.hpp"
#include <limits>
#include <memory>
#include <list>

```

Include dependency graph for Model.hpp:



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



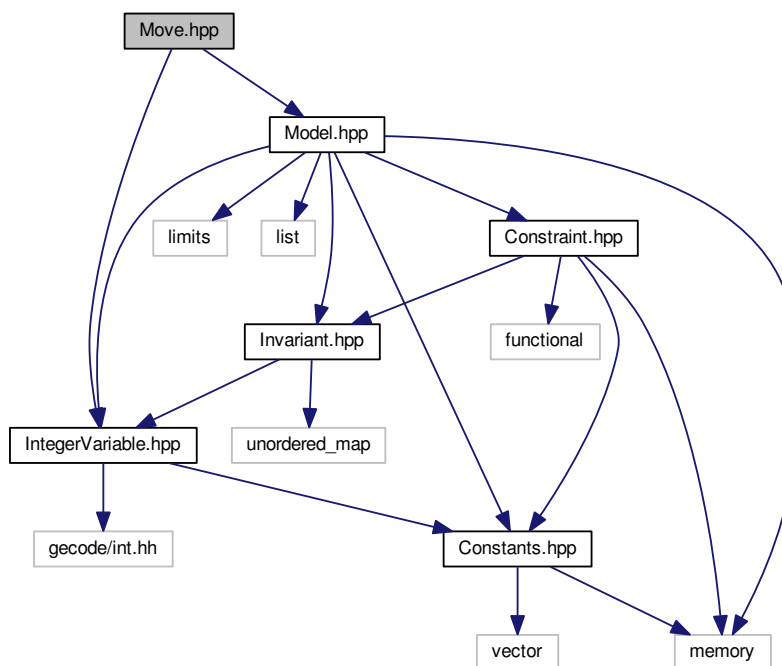
Classes

- class [Model](#)

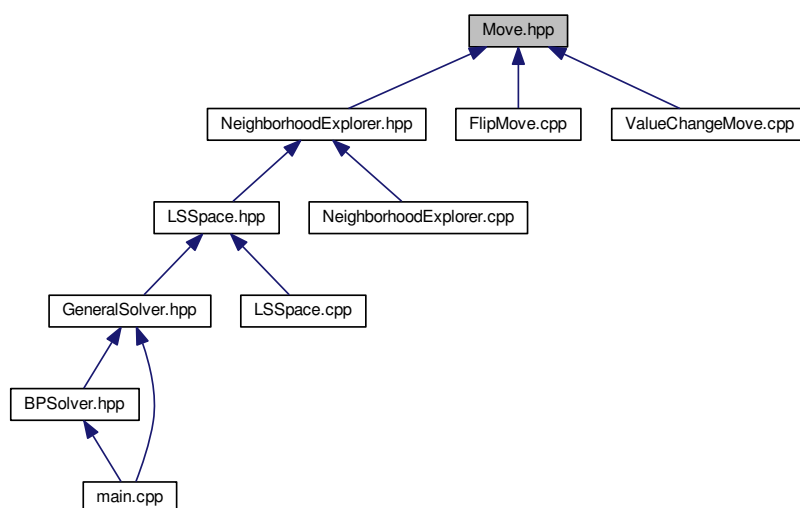
5.24 Move.hpp File Reference

```
#include "Model.hpp"
#include "IntegerVariable.hpp"
```

Include dependency graph for Move.hpp:



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



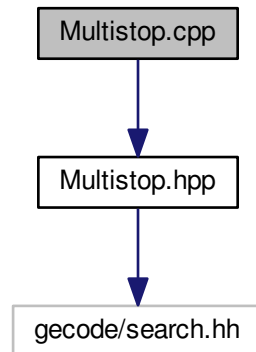
Classes

- class [Move](#)

5.25 Multistop.cpp File Reference

```
#include "Multistop.hpp"
```

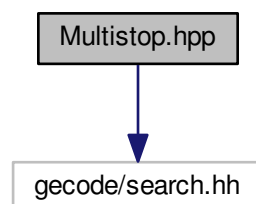
Include dependency graph for Multistop.cpp:



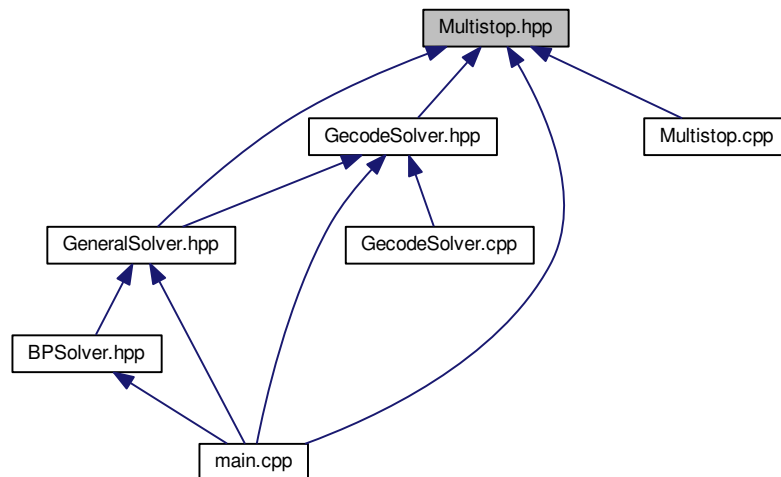
5.26 Multistop.hpp File Reference

```
#include <gecode/search.hh>
```

Include dependency graph for Multistop.hpp:



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



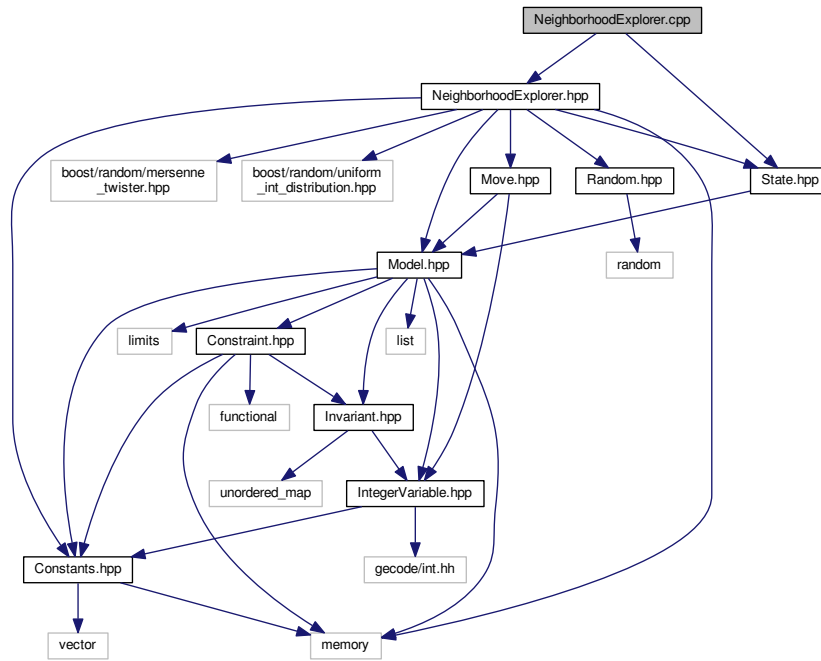
Classes

- class [Multistop](#)

5.27 NeighborhoodExplorer.cpp File Reference

```
#include "NeighborhoodExplorer.hpp"
#include "State.hpp"
```

Include dependency graph for NeighborhoodExplorer.cpp:



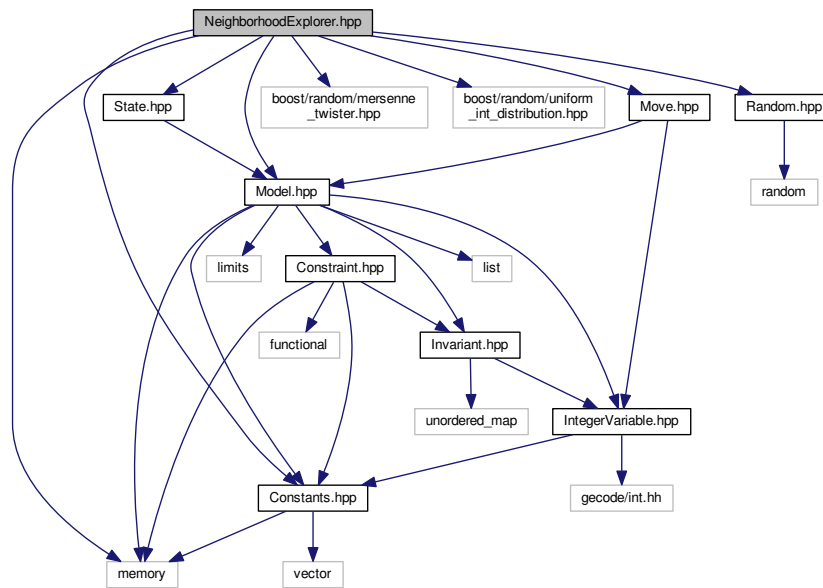
5.28 NeighborhoodExplorer.hpp File Reference

```

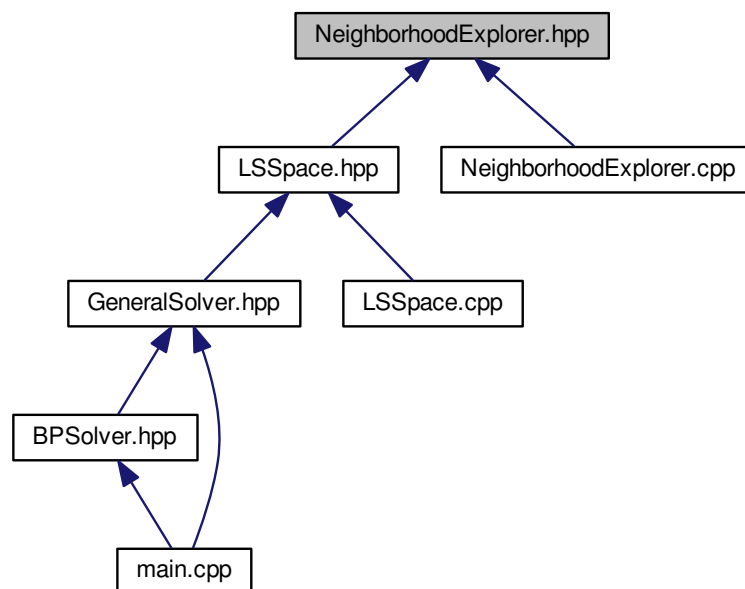
#include "Move.hpp"
#include "Model.hpp"
#include "Constants.hpp"
#include <memory>
#include "boost/random/mersenne_twister.hpp"
#include "boost/random/uniform_int_distribution.hpp"
#include "Random.hpp"
#include "State.hpp"

```


Include dependency graph for NeighborhoodExplorer.hpp:



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



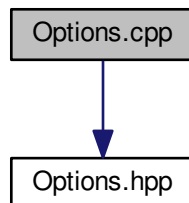
Classes

- class [NeighborhoodExplorer](#)

5.29 Options.cpp File Reference

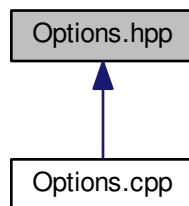
```
#include "Options.hpp"
```

Include dependency graph for Options.cpp:



5.30 Options.hpp File Reference

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



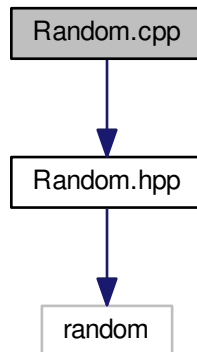
Classes

- class [Options](#)

5.31 Random.cpp File Reference

```
#include "Random.hpp"
```

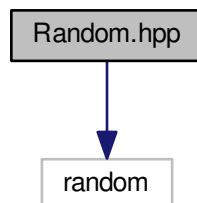
Include dependency graph for Random.cpp:



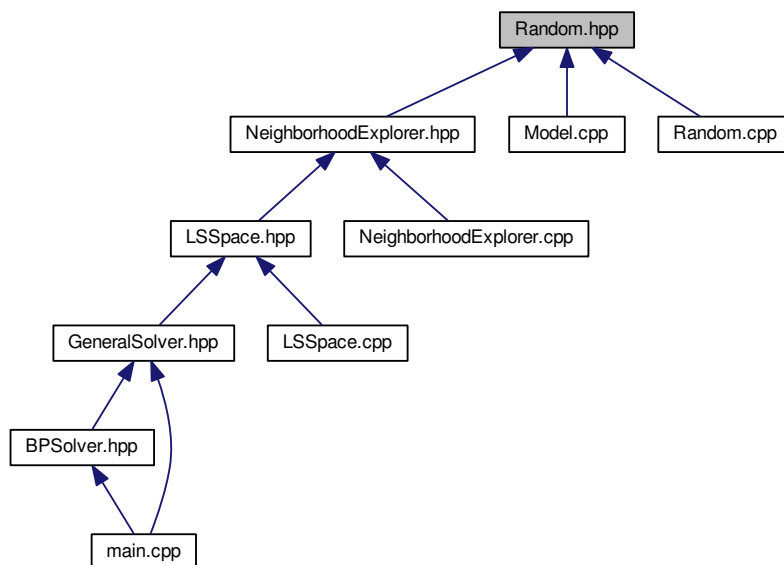
5.32 Random.hpp File Reference

```
#include <random>
```

Include dependency graph for Random.hpp:



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



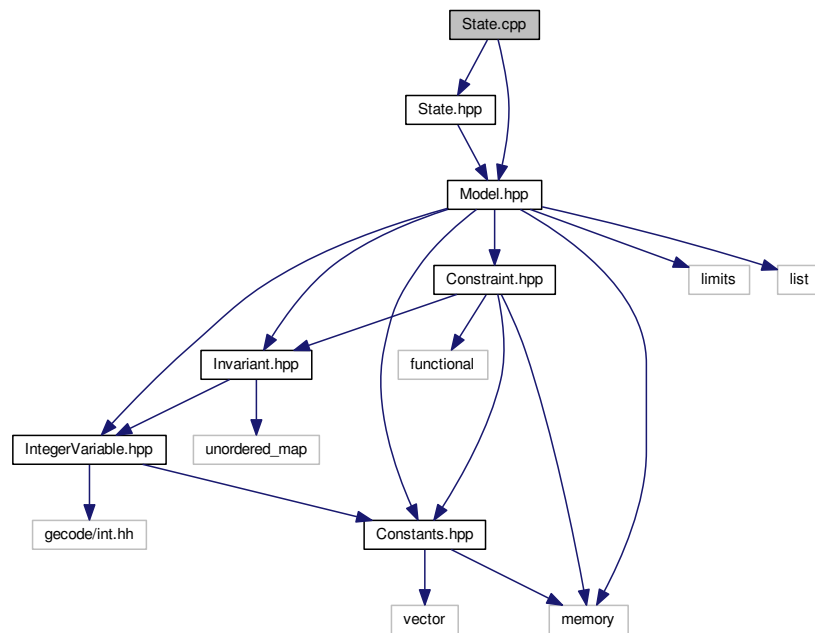
Classes

- class [Random](#)

5.33 State.cpp File Reference

```
#include "State.hpp"  
#include "Model.hpp"
```

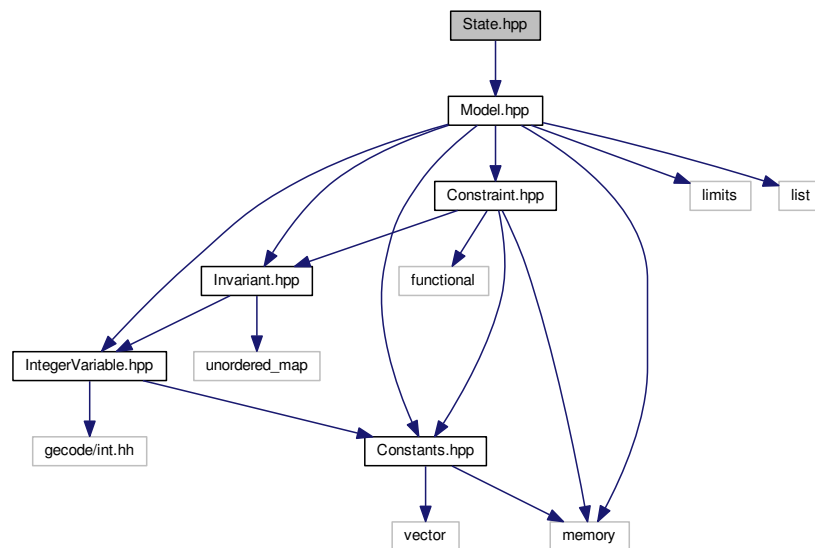
Include dependency graph for State.cpp:



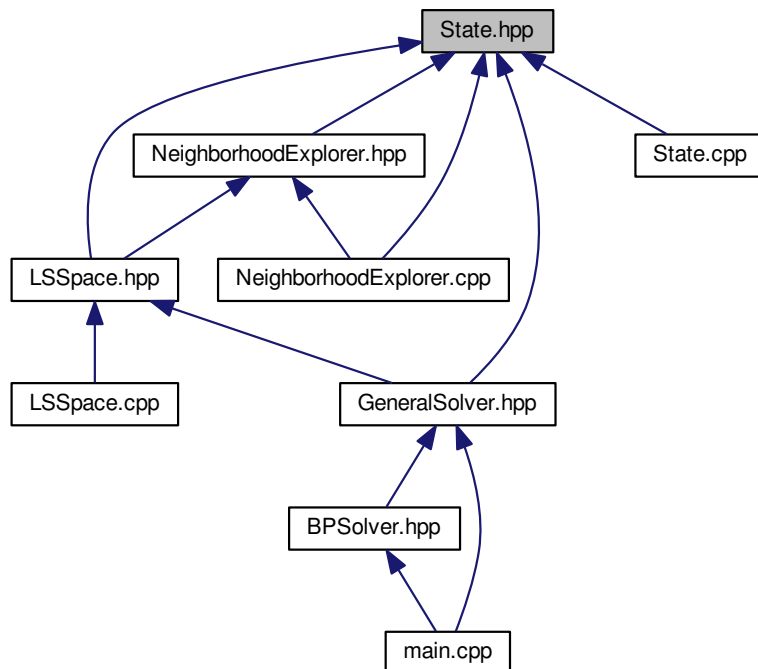
5.34 State.hpp File Reference

```
#include "Model.hpp"
```

Include dependency graph for State.hpp:



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



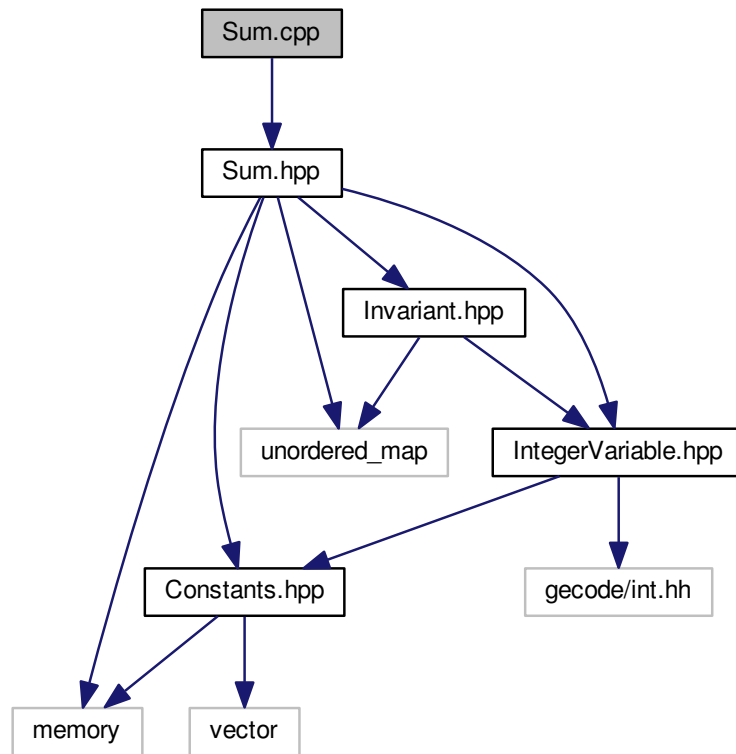
Classes

- class [State](#)

5.35 Sum.cpp File Reference

```
#include "Sum.hpp"
```

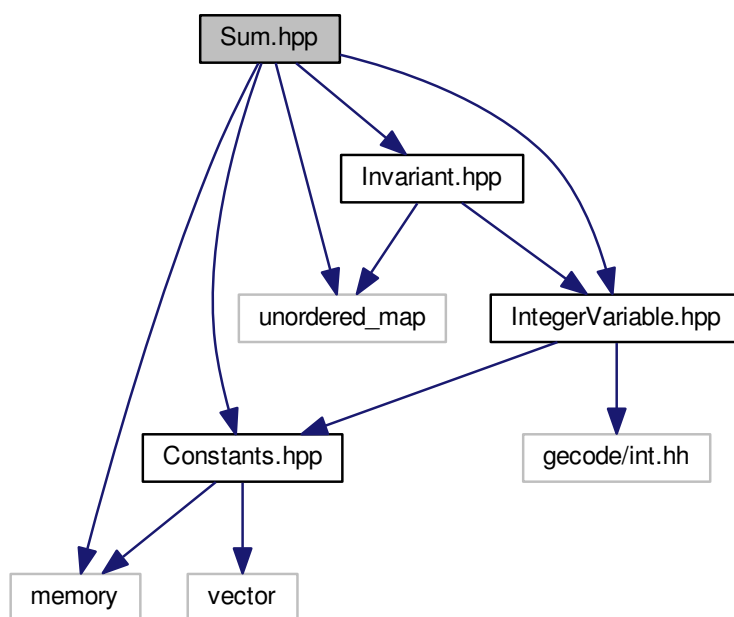
Include dependency graph for Sum.cpp:



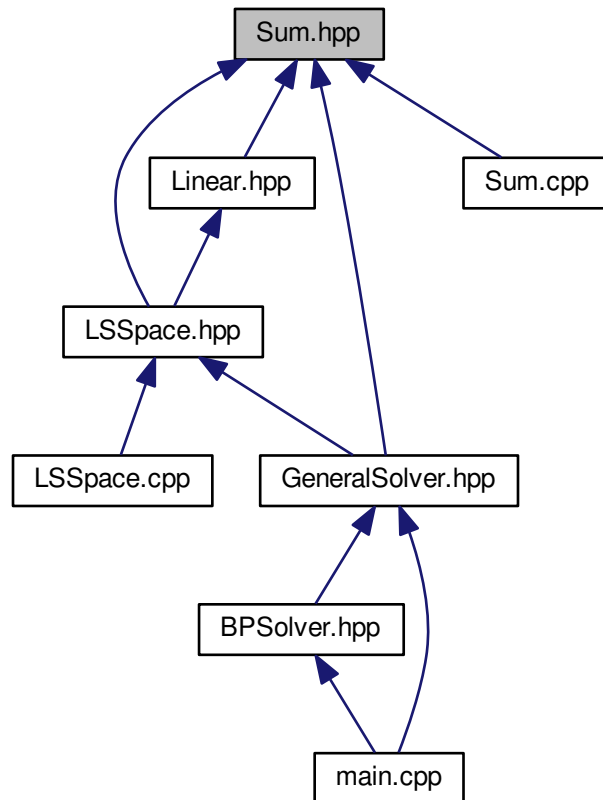
5.36 Sum.hpp File Reference

```
#include <memory>
#include "Constants.hpp"
#include <unordered_map>
#include "IntegerVariable.hpp"
#include "Invariant.hpp"
```

Include dependency graph for Sum.hpp:



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



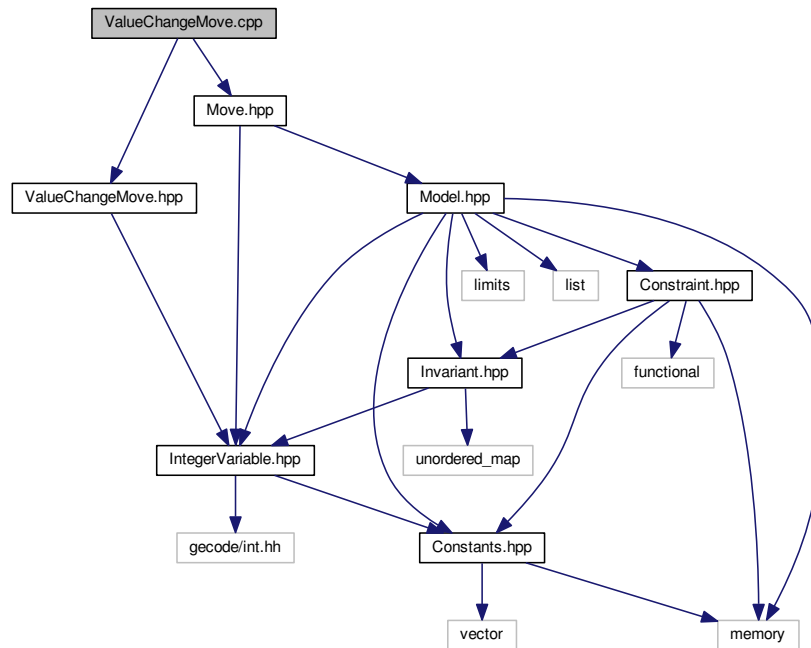
Classes

- class [Sum](#)

5.37 ValueChangeMove.cpp File Reference

```
#include "ValueChangeMove.hpp"  
#include "Move.hpp"
```

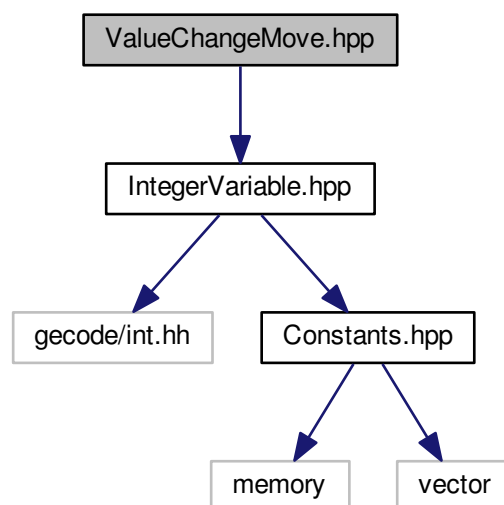
Include dependency graph for ValueChangeMove.cpp:



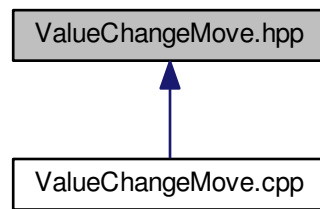
5.38 ValueChangeMove.hpp File Reference

```
#include "IntegerVariable.hpp"
```

Include dependency graph for ValueChangeMove.hpp:



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



Classes

- class [ValueChangeMove](#)

Index

- ~BPSolver
 - BPSolver, [12](#)
- ~BP_Input
 - BP_Input, [8](#)
- ~Constraint
 - Constraint, [14](#)
- ~FlipMove
 - FlipMove, [18](#)
- ~GecodeSolver
 - GecodeSolver, [19](#)
- ~GeneralSolver
 - GeneralSolver, [21](#)
- ~IntegerVariable
 - IntegerVariable, [24](#)
- ~Invariant
 - Invariant, [26](#)
- ~LSSpace
 - LSSpace, [31](#)
- ~Linear
 - Linear, [29](#)
- ~Model
 - Model, [32](#)
- ~Move
 - Move, [35](#)
- ~Multistop
 - Multistop, [37](#)
- ~NeighborhoodExplorer
 - NeighborhoodExplorer, [38](#)
- ~Options
 - Options, [39](#)
- ~State
 - State, [41](#)
- ~Sum
 - Sum, [43](#)
- ~ValueChangeMove
 - ValueChangeMove, [45](#)
- ~bounds
 - bounds, [7](#)
- ~elem
 - elem, [16](#)
- ~var
 - var, [46](#)
- addBinaryVariable
 - Model, [32](#)
- addChange
 - Invariant, [26](#)
 - Sum, [43](#)
- addIntegerVariable
 - Model, [32](#)
- addToUpdate
 - IntegerVariable, [24](#)
 - Invariant, [26](#)
- addToUsedInConstraints
 - IntegerVariable, [24](#)
- allConstraints
 - Constants.hpp, [53](#)
- arguments
 - Constraint, [15](#)
- ArrayPointer
 - IntegerVariable, [24](#)
- assign
 - BP_Output, [10](#)
- assignment
 - BP_Output, [10](#)
- BP_Data.cpp, [47](#)
 - operator<<, [47](#)
 - operator>>, [47](#)
- BP_Data.hpp, [47](#)
 - DEBUG, [48](#)
- BP_Input, [8](#)
 - ~BP_Input, [8](#)
 - BP_Input, [8](#)
 - BP_Input, [8](#)
 - bterms, [9](#)
 - direction, [9](#)
 - getBterms, [8](#)
 - getDirection, [9](#)
 - getMatcoeff, [9](#)
 - getMatcoeff2, [9](#)
 - getNcons, [9](#)
 - getNvars, [9](#)
 - getScale, [9](#)
 - getVar, [9](#)
 - getVars, [9](#)
 - matcoeff, [9](#)
 - matcoeff2, [9](#)
 - nbinvars, [9](#)
 - ncons, [9](#)
 - nintvars, [9](#)
 - nvars, [9](#)
 - operator<<, [9](#)
 - scale, [9](#)
 - setDigits, [9](#)
 - vars, [9](#)
- BP_Output, [9](#)
 - assign, [10](#)
 - assignment, [10](#)
 - BP_Output, [10](#)

- BP_Output, 10
- in, 11
- operator<<, 10
- operator>>, 10
- operator=, 10
- varAssignment, 11
- BPSolver, 11
 - ~BPSolver, 12
 - BPSolver, 12
 - BPSolver, 12
 - printCurrent, 12
- BPSolver.hpp, 48
- bestImprovement
 - LSSpace, 31
 - NeighborhoodExplorer, 38
- bin
 - var, 46
- binVars
 - GecodeSolver, 20
- bounds, 7
 - ~bounds, 7
 - lb, 7
 - type, 7
 - ub, 8
- branch
 - GecodeSolver, 19
- bterms
 - BP_Input, 9
- calculateDeltaChange
 - NeighborhoodExplorer, 38
- calculateDeltaValue
 - Invariant, 26
 - Sum, 43
- called
 - Multistop, 37
- canBeMadeOneway
 - GeneralSolver, 21
- clearUpdateVector
 - IntegerVariable, 24
- Clock, 12
 - globalClock, 12
- Clock.cpp, 49
- Clock.hpp, 50
- coefType
 - Constants.hpp, 53
- coeff
 - elem, 16
- coefficients
 - Constraint, 15
 - Invariant, 27
- commitMove
 - NeighborhoodExplorer, 38
- Constants.hpp, 50
 - allConstraints, 53
 - coefType, 53
 - constraint, 53
 - constraintContainer, 53
 - debug, 52
 - EQ, 52
 - FLIP, 52
 - GQ, 52
 - GR, 52
 - HARD, 52
 - iniTrails, 52
 - invariant, 53
 - InvariantContainer, 53
 - LE, 52
 - LINEAR, 52
 - LQ, 52
 - maxIter, 52
 - maxMoves, 52
 - nonTerm, 52
 - OBJ, 52
 - PRINT, 52
 - RANDOMSEED, 53
 - SOFT, 53
 - SUM, 53
 - SWAP, 53
 - tabulistLB, 53
 - tabulistUB, 53
 - updateType, 53
 - updateVector, 53
 - VALUECHANGE, 53
 - variableContainer, 53
 - VariableInConstraints, 53
 - weight, 53
- Constraint, 12
 - ~Constraint, 14
 - arguments, 15
 - coefficients, 15
 - Constraint, 14
 - DeltaViolation, 15
 - DeltaViolationDegree, 15
 - getArgument, 14
 - getCoefficients, 14
 - getDeltaViolation, 14
 - getDeltaViolationDegree, 14
 - getInvariant, 14
 - getNumberOfIntegerVariables, 14
 - getScopeSize, 14
 - getType, 14
 - getVariables, 14
 - getViolation, 14
 - getViolationDegree, 14
 - invariant, 15
 - isOneway, 14
 - numberOfIntegerVariables, 15
 - oneway, 15
 - operator<, 14
 - operator>, 14
 - priority, 15
 - scopeSize, 15
 - setDeltaViolation, 14
 - setDeltaViolationDegree, 14
 - setInvariant, 14
 - setNumberOfIntegerVariables, 14

- testCons, 14
- testObj, 14
- type, 15
- updateViolation, 15
- updateViolationDegree, 15
- variables, 15
- Violation, 15
- ViolationDegree, 15
- constraint
 - Constants.hpp, 53
- Constraint.hpp, 53
- Constraint::SortGreater, 40
 - operator(), 40
- constraintContainer
 - Constants.hpp, 53
- constraintPriority
 - Invariant, 27
- ConstraintSorter, 15
 - ConstraintSorter, 16
 - ConstraintSorter, 16
 - operator(), 16
- Constraints
 - Model, 33
- constraints
 - IntegerVariable, 24
- copy
 - GecodeSolver, 19
 - Move, 35
- createArray
 - GecodeSolver, 19
- createGecodeVariable
 - GecodeSolver, 19
- createIntVar
 - GeneralSolver, 21
- createIntVars
 - GeneralSolver, 21
- CurrentValue
 - Invariant, 27
- DEBUG
 - BP_Data.hpp, 48
- debug
 - Constants.hpp, 52
- definedByCons
 - IntegerVariable, 24
- DeltaValue
 - Invariant, 27
- deltaValueFirst
 - Move, 35
- deltaValueSecond
 - Move, 35
- deltaValueThird
 - Move, 35
- DeltaViolation
 - Constraint, 15
- DeltaViolationDegree
 - Constraint, 15
- direction
 - BP_Input, 9
- Double
 - Random, 39
- EQ
 - Constants.hpp, 52
- elem, 16
 - ~elem, 16
 - coeff, 16
 - index, 16
- FLIP
 - Constants.hpp, 52
- FindSolution
 - GecodeSolver, 19
- first
 - Move, 35
- fixVariables
 - GecodeSolver, 19
- flip
 - Move, 35
- FlipMove, 16
 - ~FlipMove, 18
 - FlipMove, 17
 - FlipMove, 17
 - operator=, 18
- FlipMove.cpp, 54
- FlipMove.hpp, 55
- fs
 - Multistop, 37
- GQ
 - Constants.hpp, 52
- GR
 - Constants.hpp, 52
- GS
 - GeneralSolver, 22
- GecodeSolver, 18
 - ~GecodeSolver, 19
 - binVars, 20
 - branch, 19
 - copy, 19
 - createArray, 19
 - createGecodeVariable, 19
 - FindSolution, 19
 - fixVariables, 19
 - GecodeSolver, 19
 - GecodeSolver, 19
 - initialize, 19
 - IntVars, 20
 - linear, 19
 - model, 20
 - postCovSol, 19
 - print, 19
 - print_stats, 19
 - printSpaceStatus, 20
 - SetValues, 20
 - tmpVars, 20
- GecodeSolver.cpp, 56
- GecodeSolver.hpp, 57

- GeneralSolver, 20
 - ~GeneralSolver, 21
 - canBeMadeOneway, 21
 - createIntVar, 21
 - createIntVars, 21
 - GS, 22
 - GeneralSolver, 21
 - GeneralSolver, 21
 - getAllVariables, 21
 - getInitialValue, 21
 - InitialSolution, 21
 - initializeLS, 21
 - IntegerVariable, 24
 - Invariant, 27
 - LS, 22
 - linear, 22
 - makeOneway, 22
 - model, 22
 - operator=, 22
 - optimizeSolution, 22
 - print, 22
 - print_stats, 22
 - printCurrent, 22
 - relax, 22
 - simpleRelax, 22
 - st, 22
 - Test, 22
- GeneralSolver.cpp, 58
- GeneralSolver.hpp, 58
- getAllVariables
 - GeneralSolver, 21
 - Model, 32
- getArgument
 - Constraint, 14
- getBterms
 - BP_Input, 8
- getCoefficients
 - Constraint, 14
 - Invariant, 26
- getConstraintNumber
 - Invariant, 27
- getConstraints
 - Model, 32
- getConstraintsWithPriority
 - Model, 32
- getCurrentRSS
 - getRSS.hpp, 60
- getCurrentValue
 - IntegerVariable, 24
 - Invariant, 27
- getDeltaValue
 - Invariant, 27
- getDeltaViolation
 - Constraint, 14
- getDeltaViolationDegree
 - Constraint, 14
- getDirection
 - BP_Input, 9
- getID
 - IntegerVariable, 24
- getInitialValue
 - GeneralSolver, 21
- getIntegerVariables
 - Model, 32
- getInvariant
 - Constraint, 14
- getInvariants
 - Model, 32
- getLowerBound
 - IntegerVariable, 24
- getMask
 - Model, 33
- getMaskAt
 - Model, 33
- getMatcoeff
 - BP_Input, 9
- getMatcoeff2
 - BP_Input, 9
- getNcons
 - BP_Input, 9
- getNonFixedBinaryVariable
 - Model, 33
- getNonFixedBinaryVariables
 - Model, 33
- getNumberOfIntegerVariables
 - Constraint, 14
- getNvars
 - BP_Input, 9
- getObjectiveValue
 - State, 41
- getObjectives
 - Model, 33
- getOneway
 - IntegerVariable, 24
- getPeakRSS
 - getRSS.hpp, 60
- getPriority
 - Invariant, 27
- getRSS.hpp, 60
 - getCurrentRSS, 60
 - getPeakRSS, 60
- getScale
 - BP_Input, 9
- getScopeSize
 - Constraint, 14
- getSolution
 - State, 41
- getSolutionValue
 - State, 41
- getType
 - Constraint, 14
 - Invariant, 27
- getUpdateVector
 - IntegerVariable, 24
 - Invariant, 27
- getUpperBound

- IntegerVariable, 24
- getUsedInObjective
 - Invariant, 27
- getVar
 - BP_Input, 9
- getVariableID
 - Invariant, 27
- getVariablePointer
 - IntegerVariable, 24
- getVariables
 - Constraint, 14
 - Invariant, 27
- getVars
 - BP_Input, 9
- getViolation
 - Constraint, 14
- getViolationDegree
 - Constraint, 14
- globalClock
 - Clock, 12
- HARD
 - Constants.hpp, 52
- in
 - BP_Output, 11
- index
 - elem, 16
- iniTrails
 - Constants.hpp, 52
- InitialSolution
 - GeneralSolver, 21
- initialValue
 - Model, 33
- initialize
 - GecodeSolver, 19
 - Invariant, 27
- initializeConstraints
 - State, 41
- initializeInvariants
 - State, 41
- initializeLS
 - GeneralSolver, 21
- initializeObjective
 - State, 41
- IntVars
 - GecodeSolver, 20
- Integer
 - Random, 39
- IntegerVariable, 22
 - ~IntegerVariable, 24
 - addToUpdate, 24
 - addToUsedInConstraints, 24
 - ArrayPointer, 24
 - clearUpdateVector, 24
 - constraints, 24
 - definedByCons, 24
 - GeneralSolver, 24
 - getCurrentValue, 24
 - getID, 24
 - getLowerBound, 24
 - getOneway, 24
 - getUpdateVector, 24
 - getUpperBound, 24
 - getVariablePointer, 24
 - IntegerVariable, 24
 - IntegerVariable, 24
 - isDefined, 24
 - isInteger, 24
 - isIntegerVariable, 24
 - lowerBound, 24
 - oneway, 24
 - setCurrentValue, 24
 - setDefinedBy, 24
 - setVariablePointer, 24
 - update, 25
 - upperBound, 25
 - usedIn, 25
 - usedInConstraints, 24
 - value, 25
 - VariablePointer, 25
 - vectorID, 25
- IntegerVariable.cpp, 60
- IntegerVariable.hpp, 60
- IntegerVariables
 - Model, 33
- Invariant, 25
 - ~Invariant, 26
 - addChange, 26
 - addToUpdate, 26
 - calculateDeltaValue, 26
 - coefficients, 27
 - constraintPriority, 27
 - CurrentValue, 27
 - DeltaValue, 27
 - GeneralSolver, 27
 - getCoefficients, 26
 - getConstraintNumber, 27
 - getCurrentValue, 27
 - getDeltaValue, 27
 - getPriority, 27
 - getType, 27
 - getUpdateVector, 27
 - getUsedInObjective, 27
 - getVariableID, 27
 - getVariables, 27
 - initialize, 27
 - Invariant, 26
 - invariants, 27
 - setUsedByConstraint, 27
 - setUsedByObjective, 27
 - startValue, 27
 - test, 27
 - type, 28
 - update, 28
 - updateValue, 27
 - usedInConstraintNr, 28

- usedInObjectiveNr, 28
 - variableID, 28
 - VariablePointers, 28
- invariant
 - Constants.hpp, 53
 - Constraint, 15
- Invariant.hpp, 61
- InvariantContainer
 - Constants.hpp, 53
- Invariants
 - Model, 33
- invariants
 - Invariant, 27
- isDefined
 - IntegerVariable, 24
- isInteger
 - IntegerVariable, 24
- isIntegerVariable
 - IntegerVariable, 24
- isOneway
 - Constraint, 14
- iterations
 - LSSpace, 31
- kind
 - var, 46
- LE
 - Constants.hpp, 52
- LINEAR
 - Constants.hpp, 52
- LQ
 - Constants.hpp, 52
- LS
 - GeneralSolver, 22
- LSSpace, 30
 - ~LSSpace, 31
 - bestImprovement, 31
 - iterations, 31
 - LSSpace, 31
 - LSSpace, 31
 - model, 31
 - optimizeSolution, 31
 - printCurrent, 31
 - simpleMove, 31
 - Test, 31
 - Violations, 31
- LSSpace.cpp, 64
- LSSpace.hpp, 64
- lb
 - bounds, 7
- Linear, 28
 - ~Linear, 29
 - Linear, 29
 - relation, 30
 - rhs, 30
 - setDeltaViolation, 29
 - setDeltaViolationDegree, 29
 - testCons, 30
 - testObj, 30
 - updateViolation, 30
 - updateViolationDegree, 30
- linear
 - GecodeSolver, 19
 - GeneralSolver, 22
- Linear.hpp, 62
- lowerBound
 - IntegerVariable, 24
- main
 - main.cpp, 66
- main.cpp, 65
 - main, 66
- makeMove
 - NeighborhoodExplorer, 38
- makeOneway
 - GeneralSolver, 22
- mask
 - Model, 33
 - State, 41
- maskAt
 - State, 41
- matcoeff
 - BP_Input, 9
- matcoeff2
 - BP_Input, 9
- maxIter
 - Constants.hpp, 52
- maxMoves
 - Constants.hpp, 52
- Model, 31
 - ~Model, 32
 - addBinaryVariable, 32
 - addIntegerVariable, 32
 - Constraints, 33
 - getAllVariables, 32
 - getConstraints, 32
 - getConstraintsWithPriority, 32
 - getIntegerVariables, 32
 - getInvariants, 32
 - getMask, 33
 - getMaskAt, 33
 - getNonFixedBinaryVariable, 33
 - getNonFixedBinaryVariables, 33
 - getObjectives, 33
 - initialValue, 33
 - IntegerVariables, 33
 - Invariants, 33
 - mask, 33
 - Model, 32
 - nonFixedBinaryVariables, 33
 - nonFixedVariables, 33
 - original, 33
 - updateIntegerVariable, 33
- model
 - GecodeSolver, 20
 - GeneralSolver, 22
 - LSSpace, 31

- NeighborhoodExplorer, 38
- State, 41
- Model.cpp, 66
- Model.hpp, 67
- Move, 34
 - ~Move, 35
 - copy, 35
 - deltaValueFirst, 35
 - deltaValueSecond, 35
 - deltaValueThird, 35
 - first, 35
 - flip, 35
 - Move, 35
 - moveType, 35
 - operator=, 35
 - second, 35
 - third, 35
- Move.hpp, 68
- moveType
 - Move, 35
- mt
 - Random, 39
- Multistop, 35
 - ~Multistop, 37
 - called, 37
 - fs, 37
 - Multistop, 37
 - ns, 37
 - stop, 37
 - ts, 37
- Multistop.cpp, 70
- Multistop.hpp, 70
- nbinvars
 - BP_Input, 9
- ncons
 - BP_Input, 9
- NeighborhoodExplorer, 37
 - ~NeighborhoodExplorer, 38
 - bestImprovement, 38
 - calculateDeltaChange, 38
 - commitMove, 38
 - makeMove, 38
 - model, 38
 - NeighborhoodExplorer, 38
 - NeighborhoodExplorer, 38
 - randomWalk, 38
- NeighborhoodExplorer.cpp, 71
- NeighborhoodExplorer.hpp, 72
- nintvars
 - BP_Input, 9
- nonFixedBinaryVariables
 - Model, 33
- nonFixedVariables
 - Model, 33
- nonTerm
 - Constants.hpp, 52
- ns
 - Multistop, 37
- numberOfIntegerVariables
 - Constraint, 15
- numberOfViolations
 - State, 42
- nvars
 - BP_Input, 9
- OBJ
 - Constants.hpp, 52
- objcoeff
 - var, 46
- oneway
 - Constraint, 15
 - IntegerVariable, 24
- operator<
 - Constraint, 14
- operator<<
 - BP_Data.cpp, 47
 - BP_Input, 9
 - BP_Output, 10
- operator>
 - Constraint, 14
- operator>>
 - BP_Data.cpp, 47
 - BP_Output, 10
- operator()
 - Constraint::SortGreater, 40
 - ConstraintSorter, 16
- operator=
 - BP_Output, 10
 - FlipMove, 18
 - GeneralSolver, 22
 - Move, 35
 - Sum, 43
 - ValueChangeMove, 45
- optimizeSolution
 - GeneralSolver, 22
 - LSSpace, 31
- Options, 38
 - ~Options, 39
 - Options, 39
 - timelimit, 39
- Options.cpp, 74
- Options.hpp, 74
- original
 - Model, 33
- PRINT
 - Constants.hpp, 52
- postCovSol
 - GecodeSolver, 19
- print
 - GecodeSolver, 19
 - GeneralSolver, 22
- print_stats
 - GecodeSolver, 19
 - GeneralSolver, 22
- printCurrent
 - BPSolver, 12

- GeneralSolver, 22
- LSSpace, 31
- printSpaceStatus
 - GecodeSolver, 20
- priority
 - Constraint, 15
- RANDOMSEED
 - Constants.hpp, 53
- Random, 39
 - Double, 39
 - Integer, 39
 - mt, 39
 - Seed, 39
 - seed, 39
- Random.cpp, 75
- Random.hpp, 75
- randomWalk
 - NeighborhoodExplorer, 38
- recalculateAll
 - State, 41
- relation
 - Linear, 30
- relax
 - GeneralSolver, 22
- rhs
 - Linear, 30
- SOFT
 - Constants.hpp, 53
- SUM
 - Constants.hpp, 53
- SWAP
 - Constants.hpp, 53
- saveSolution
 - State, 41
- scale
 - BP_Input, 9
- scopeSize
 - Constraint, 15
- second
 - Move, 35
- Seed
 - Random, 39
- seed
 - Random, 39
- setCurrentValue
 - IntegerVariable, 24
- setDefinedBy
 - IntegerVariable, 24
- setDeltaViolation
 - Constraint, 14
 - Linear, 29
- setDeltaViolationDegree
 - Constraint, 14
 - Linear, 29
- setDigits
 - BP_Input, 9
- setInvariant
 - Constraint, 14
- setNumberOfIntegerVariables
 - Constraint, 14
- setSolution
 - State, 41
- setUsedByConstraint
 - Invariant, 27
- setUsedByObjective
 - Invariant, 27
- SetValues
 - GecodeSolver, 20
- setVariablePointer
 - IntegerVariable, 24
- shuffleMask
 - State, 41
- simpleMove
 - LSSpace, 31
- simpleRelax
 - GeneralSolver, 22
- solution
 - State, 42
- solutionValue
 - State, 42
- st
 - GeneralSolver, 22
- startValue
 - Invariant, 27
- State, 40
 - ~State, 41
 - getObjectiveValue, 41
 - getSolution, 41
 - getSolutionValue, 41
 - initializeConstraints, 41
 - initializeInvariants, 41
 - initializeObjective, 41
 - mask, 41
 - maskAt, 41
 - model, 41
 - numberOfViolations, 42
 - recalculateAll, 41
 - saveSolution, 41
 - setSolution, 41
 - shuffleMask, 41
 - solution, 42
 - solutionValue, 42
 - State, 41
- State.cpp, 76
- State.hpp, 77
- stop
 - Multistop, 37
- Sum, 42
 - ~Sum, 43
 - addChange, 43
 - calculateDeltaValue, 43
 - operator=, 43
 - Sum, 43
 - test, 43
 - VariableChange, 43

- Sum.cpp, 78
- Sum.hpp, 79
- tabulistLB
 - Constants.hpp, 53
- tabulistUB
 - Constants.hpp, 53
- Test
 - GeneralSolver, 22
 - LSSpace, 31
- test
 - Invariant, 27
 - Sum, 43
- testCons
 - Constraint, 14
 - Linear, 30
- testObj
 - Constraint, 14
 - Linear, 30
- third
 - Move, 35
- timelimit
 - Options, 39
- tmpVars
 - GecodeSolver, 20
- ts
 - Multistop, 37
- type
 - bounds, 7
 - Constraint, 15
 - Invariant, 28
- ub
 - bounds, 8
- update
 - IntegerVariable, 25
 - Invariant, 28
- updateIntegerVariable
 - Model, 33
- updateType
 - Constants.hpp, 53
- updateValue
 - Invariant, 27
- updateVector
 - Constants.hpp, 53
- updateViolation
 - Constraint, 15
 - Linear, 30
- updateViolationDegree
 - Constraint, 15
 - Linear, 30
- upperBound
 - IntegerVariable, 25
- usedIn
 - IntegerVariable, 25
- usedInConstraintNr
 - Invariant, 28
- usedInConstraints
 - IntegerVariable, 24
- usedInObjectiveNr
 - Invariant, 28
- VALUECHANGE
 - Constants.hpp, 53
- value
 - IntegerVariable, 25
- ValueChangeMove, 44
 - ~ValueChangeMove, 45
 - operator=, 45
 - ValueChangeMove, 44, 45
 - ValueChangeMove, 44, 45
- ValueChangeMove.cpp, 81
- ValueChangeMove.hpp, 82
- var, 45
 - ~var, 46
 - bin, 46
 - kind, 46
 - objcoeff, 46
- varAssignment
 - BP_Output, 11
- VariableChange
 - Sum, 43
- variableContainer
 - Constants.hpp, 53
- variableID
 - Invariant, 28
- VariableInConstraints
 - Constants.hpp, 53
- VariablePointer
 - IntegerVariable, 25
- VariablePointers
 - Invariant, 28
- variables
 - Constraint, 15
- vars
 - BP_Input, 9
- vectorID
 - IntegerVariable, 25
- Violation
 - Constraint, 15
- ViolationDegree
 - Constraint, 15
- Violations
 - LSSpace, 31
- weight
 - Constants.hpp, 53