

# IEGENLIB

Generated by Doxygen 1.7.5

Fri Sep 21 2012 09:21:17



# Contents

- 1 IEGen C++ Rewrite** **1**
  - 1.1 Intro . . . . . 1
  - 1.2 COPYING . . . . . 2
  - 1.3 README . . . . . 2
  
- 2 Directory Hierarchy** **9**
  - 2.1 Directories . . . . . 9
  
- 3 Namespace Index** **11**
  - 3.1 Namespace List . . . . . 11
  
- 4 Class Index** **13**
  - 4.1 Class Hierarchy . . . . . 13
  
- 5 Class Index** **15**
  - 5.1 Class List . . . . . 15
  
- 6 File Index** **17**
  - 6.1 File List . . . . . 17
  
- 7 Directory Documentation** **19**
  - 7.1 src/iegenlib/ Directory Reference . . . . . 20
  - 7.2 src/iegenlib/parser/ Directory Reference . . . . . 21
  - 7.3 src/iegenlib/set\_relation/ Directory Reference . . . . . 22
  - 7.4 src/ Directory Reference . . . . . 23

---

<b>8 Namespace Documentation</b>	<b>25</b>
8.1 iegenlib Namespace Reference	25
8.1.1 Detailed Description	26
8.1.2 Function Documentation	26
8.1.2.1 appendCurrEnv	26
8.1.2.2 queryInverseCurrEnv	27
8.1.2.3 setCurrEnv	28
8.1.2.4 setCurrEnv	29
8.1.3 Variable Documentation	29
8.1.3.1 currentEnv	29
8.2 iegenlib::parser Namespace Reference	29
8.2.1 Function Documentation	30
8.2.1.1 clearAll	30
8.2.1.2 get_parse_env_result	31
8.2.1.3 get_parse_error	31
8.2.1.4 get_parse_relation_result	31
8.2.1.5 get_parse_set_result	31
8.2.1.6 parse_env	31
8.2.1.7 parse_relation	32
8.2.1.8 parse_relation	33
8.2.1.9 parse_set	34
8.2.1.10 parse_set	34
8.2.1.11 set_parse_env_result	35
8.2.1.12 set_parse_error	36
8.2.1.13 set_parse_error	36
8.2.1.14 set_parse_relation_result	37
8.2.1.15 set_parse_set_result	37
8.2.1.16 string_get_next_char	38
8.2.2 Variable Documentation	38
8.2.2.1 error_message	38
8.2.2.2 input_buffer	38

---

8.2.2.3	input_pos . . . . .	38
8.2.2.4	parse_env_result . . . . .	39
8.2.2.5	parse_error . . . . .	39
8.2.2.6	parse_relation_result . . . . .	39
8.2.2.7	parse_set_result . . . . .	39
<b>9</b>	<b>Class Documentation</b>	<b>41</b>
9.1	iegenlib::Conjunction Class Reference . . . . .	41
9.1.1	Detailed Description . . . . .	42
9.1.2	Constructor & Destructor Documentation . . . . .	43
9.1.2.1	Conjunction . . . . .	43
9.1.2.2	Conjunction . . . . .	43
9.1.2.3	Conjunction . . . . .	43
9.1.2.4	~Conjunction . . . . .	43
9.1.3	Member Function Documentation . . . . .	43
9.1.3.1	addEquality . . . . .	43
9.1.3.2	addInequality . . . . .	44
9.1.3.3	Apply . . . . .	45
9.1.3.4	arity . . . . .	46
9.1.3.5	cleanUp . . . . .	47
9.1.3.6	clone . . . . .	47
9.1.3.7	Compose . . . . .	48
9.1.3.8	copyConstraintsFrom . . . . .	49
9.1.3.9	equalities . . . . .	50
9.1.3.10	findAndRemoveFunction . . . . .	50
9.1.3.11	findFunction . . . . .	51
9.1.3.12	inarity . . . . .	52
9.1.3.13	inequalities . . . . .	52
9.1.3.14	Inverse . . . . .	52
9.1.3.15	isFunction . . . . .	53
9.1.3.16	isFunctionInverse . . . . .	54

---

9.1.3.17	operator= . . . . .	54
9.1.3.18	prettyPrintString . . . . .	54
9.1.3.19	remapTupleVars . . . . .	55
9.1.3.20	satisfiable . . . . .	55
9.1.3.21	setTupleElem . . . . .	56
9.1.3.22	setTupleElem . . . . .	56
9.1.3.23	substituteInConstraints . . . . .	56
9.1.3.24	substituteTupleDecl . . . . .	57
9.1.3.25	toDotString . . . . .	58
9.1.3.26	toDotString . . . . .	59
9.1.3.27	toString . . . . .	59
9.1.4	Member Data Documentation . . . . .	60
9.1.4.1	mEqualities . . . . .	60
9.1.4.2	mInArity . . . . .	60
9.1.4.3	mInequalities . . . . .	60
9.1.4.4	mTupleDecl . . . . .	60
9.2	iegenlib::Environment Class Reference . . . . .	60
9.2.1	Constructor & Destructor Documentation . . . . .	61
9.2.1.1	Environment . . . . .	61
9.2.1.2	Environment . . . . .	61
9.2.1.3	Environment . . . . .	62
9.2.2	Member Function Documentation . . . . .	62
9.2.2.1	append . . . . .	62
9.2.2.2	funclInverse . . . . .	62
9.2.2.3	hasInverse . . . . .	63
9.2.2.4	operator= . . . . .	63
9.2.2.5	reset . . . . .	64
9.2.2.6	setlInverse . . . . .	64
9.2.2.7	toString . . . . .	65
9.2.3	Member Data Documentation . . . . .	65
9.2.3.1	mInverseMap . . . . .	65

---

---

9.3	EnvironmentTest Class Reference	65
9.3.1	Member Function Documentation	65
9.3.1.1	SetUp	65
9.3.1.2	TearDown	65
9.4	ExceptionTestExp Class Reference	66
9.4.1	Detailed Description	66
9.4.2	Member Function Documentation	66
9.4.2.1	SetUp	66
9.4.2.2	TearDown	66
9.5	ExceptionTestParser Class Reference	66
9.5.1	Detailed Description	66
9.5.2	Member Function Documentation	67
9.5.2.1	SetUp	67
9.5.2.2	TearDown	67
9.6	ExceptionTestSetRelation Class Reference	67
9.6.1	Detailed Description	67
9.6.2	Member Function Documentation	67
9.6.2.1	SetUp	67
9.6.2.2	TearDown	67
9.7	iegenlib::Exp Class Reference	67
9.7.1	Detailed Description	69
9.7.2	Member Enumeration Documentation	70
9.7.2.1	exptype	70
9.7.3	Constructor & Destructor Documentation	70
9.7.3.1	Exp	70
9.7.3.2	Exp	70
9.7.3.3	~Exp	70
9.7.4	Member Function Documentation	70
9.7.4.1	addExp	71
9.7.4.2	addTerm	71
9.7.4.3	clone	72

---

9.7.4.4	<a href="#">collapseNestedInvertibleFunctions</a>	73
9.7.4.5	<a href="#">dependsOn</a>	74
9.7.4.6	<a href="#">divideBy</a>	74
9.7.4.7	<a href="#">equalsZero</a>	74
9.7.4.8	<a href="#">findMatchingFactor</a>	75
9.7.4.9	<a href="#">getExpType</a>	76
9.7.4.10	<a href="#">getTerm</a>	76
9.7.4.11	<a href="#">isConst</a>	76
9.7.4.12	<a href="#">isDivisible</a>	77
9.7.4.13	<a href="#">isEquality</a>	78
9.7.4.14	<a href="#">isExpression</a>	78
9.7.4.15	<a href="#">isInequality</a>	78
9.7.4.16	<a href="#">multiplyBy</a>	79
9.7.4.17	<a href="#">normalizeForEquality</a>	79
9.7.4.18	<a href="#">operator=</a>	80
9.7.4.19	<a href="#">operator==</a>	80
9.7.4.20	<a href="#">prettyPrintString</a>	81
9.7.4.21	<a href="#">remapTupleVars</a>	81
9.7.4.22	<a href="#">setEquality</a>	82
9.7.4.23	<a href="#">setExpression</a>	82
9.7.4.24	<a href="#">setInequality</a>	82
9.7.4.25	<a href="#">solveForFactor</a>	82
9.7.4.26	<a href="#">substitute</a>	83
9.7.4.27	<a href="#">toDotString</a>	84
9.7.4.28	<a href="#">toDotString</a>	84
9.7.4.29	<a href="#">toString</a>	85
9.7.5	<a href="#">Member Data Documentation</a>	85
9.7.5.1	<a href="#">mExpType</a>	85
9.7.5.2	<a href="#">mTerms</a>	85
9.8	<a href="#">ExpTest Class Reference</a>	85
9.8.1	<a href="#">Member Function Documentation</a>	86

---



---

9.8.1.1	Setup	86
9.8.1.2	TearDown	86
9.8.2	Member Data Documentation	86
9.8.2.1	dot_data	86
9.8.2.2	root_dir	86
9.9	iegenlib::Relation Class Reference	86
9.9.1	Detailed Description	89
9.9.2	Constructor & Destructor Documentation	90
9.9.2.1	Relation	90
9.9.2.2	Relation	90
9.9.2.3	Relation	90
9.9.2.4	~Relation	90
9.9.3	Member Function Documentation	90
9.9.3.1	addConjunction	91
9.9.3.2	Apply	92
9.9.3.3	arity	92
9.9.3.4	Compose	93
9.9.3.5	inArity	94
9.9.3.6	Inverse	94
9.9.3.7	operator=	95
9.9.3.8	prettyPrintString	95
9.9.3.9	remapTupleVars	96
9.9.3.10	toDotString	96
9.9.3.11	toString	96
9.9.3.12	Union	97
9.9.4	Member Data Documentation	98
9.9.4.1	mInArity	98
9.9.4.2	mOutArity	98
9.10	iegenlib::Set Class Reference	98
9.10.1	Detailed Description	101
9.10.2	Constructor & Destructor Documentation	101

9.10.2.1	Set	101
9.10.2.2	Set	102
9.10.2.3	Set	102
9.10.2.4	~Set	102
9.10.3	Member Function Documentation	102
9.10.3.1	arity	102
9.10.3.2	operator=	102
9.10.3.3	toDotString	102
9.10.3.4	Union	103
9.10.4	Member Data Documentation	104
9.10.4.1	mArity	104
9.11	SetRelationTest Class Reference	105
9.11.1	Member Function Documentation	106
9.11.1.1	SetUp	106
9.11.1.2	TearDown	107
9.11.2	Member Data Documentation	107
9.11.2.1	conj1	107
9.11.2.2	conj1r	107
9.11.2.3	conj2	107
9.11.2.4	conj2r	107
9.11.2.5	dot_data	107
9.11.2.6	e1	107
9.11.2.7	e2	107
9.11.2.8	negx_plus_y	107
9.11.2.9	root_dir	107
9.11.2.10	x	107
9.11.2.11	x_minus_y	107
9.12	iegenlib::SparseConstraints Class Reference	107
9.12.1	Detailed Description	109
9.12.2	Constructor & Destructor Documentation	109
9.12.2.1	SparseConstraints	109

---

9.12.2.2	SparseConstraints	109
9.12.2.3	~SparseConstraints	110
9.12.3	Member Function Documentation	110
9.12.3.1	addConjunction	110
9.12.3.2	arity	110
9.12.3.3	cleanUp	111
9.12.3.4	conjunctionBegin	111
9.12.3.5	conjunctionEnd	112
9.12.3.6	operator=	112
9.12.3.7	prettyPrintString	112
9.12.3.8	prettyPrintString	113
9.12.3.9	toDotString	114
9.12.3.10	toString	114
9.12.3.11	toString	115
9.12.4	Member Data Documentation	115
9.12.4.1	mConjunctions	115
9.13	iegenlib::Term Class Reference	115
9.13.1	Detailed Description	118
9.13.2	Member Enumeration Documentation	118
9.13.2.1	termtype	118
9.13.3	Constructor & Destructor Documentation	118
9.13.3.1	Term	118
9.13.3.2	Term	119
9.13.3.3	~Term	119
9.13.4	Member Function Documentation	119
9.13.4.1	clone	119
9.13.4.2	coefficient	120
9.13.4.3	coeffToStream	120
9.13.4.4	collapseNestedInvertibleFunctions	121
9.13.4.5	combine	121
9.13.4.6	compareTermTypes	122

---

9.13.4.7	divideBy	123
9.13.4.8	factorMatches	123
9.13.4.9	getExpList	124
9.13.4.10	getTermType	124
9.13.4.11	isConst	124
9.13.4.12	isUFCall	125
9.13.4.13	multiplyBy	125
9.13.4.14	operator<	125
9.13.4.15	operator=	126
9.13.4.16	operator==	126
9.13.4.17	prettyPrintString	126
9.13.4.18	setTermType	126
9.13.4.19	toDotString	126
9.13.4.20	toString	127
9.13.4.21	type	127
9.13.5	Member Data Documentation	128
9.13.5.1	mCoeff	128
9.13.5.2	mTermType	128
9.14	iegenlib::TupleElemDecl Class Reference	128
9.14.1	Detailed Description	129
9.14.2	Constructor & Destructor Documentation	129
9.14.2.1	TupleElemDecl	129
9.14.2.2	TupleElemDecl	129
9.14.2.3	TupleElemDecl	129
9.14.2.4	TupleElemDecl	129
9.14.3	Member Function Documentation	129
9.14.3.1	constVal	129
9.14.3.2	createTerm	129
9.14.3.3	isConst	130
9.14.3.4	operator=	130
9.14.3.5	toString	130

---

9.14.3.6	tupleDeclToString	131
9.14.3.7	varString	131
9.14.4	Member Data Documentation	131
9.14.4.1	mConstVal	131
9.14.4.2	mIsConst	131
9.14.4.3	mVarString	131
9.15	iegenlib::TupleVarTerm Class Reference	131
9.15.1	Detailed Description	134
9.15.2	Constructor & Destructor Documentation	135
9.15.2.1	TupleVarTerm	135
9.15.2.2	TupleVarTerm	135
9.15.2.3	TupleVarTerm	136
9.15.3	Member Function Documentation	136
9.15.3.1	clone	136
9.15.3.2	factorMatches	136
9.15.3.3	isConst	137
9.15.3.4	operator<	137
9.15.3.5	operator=	137
9.15.3.6	prettyPrintString	137
9.15.3.7	remapLocation	138
9.15.3.8	toString	139
9.15.3.9	type	139
9.15.4	Member Data Documentation	140
9.15.4.1	mLocation	140
9.16	iegenlib::UFCallTerm Class Reference	140
9.16.1	Detailed Description	144
9.16.2	Constructor & Destructor Documentation	144
9.16.2.1	UFCallTerm	144
9.16.2.2	UFCallTerm	145
9.16.2.3	UFCallTerm	145
9.16.2.4	~UFCallTerm	145

---

9.16.3	Member Function Documentation	145
9.16.3.1	argsToStream	145
9.16.3.2	argsToStreamPrettyPrint	146
9.16.3.3	clone	146
9.16.3.4	collapseNestedInvertibleFunctions	146
9.16.3.5	factorMatches	147
9.16.3.6	getExpList	147
9.16.3.7	isConst	148
9.16.3.8	isUFCall	148
9.16.3.9	name	148
9.16.3.10	operator<	148
9.16.3.11	operator=	149
9.16.3.12	prettyPrintString	149
9.16.3.13	toDotString	149
9.16.3.14	toString	150
9.16.3.15	type	151
9.16.4	Member Data Documentation	151
9.16.4.1	mArgs	151
9.16.4.2	mFuncName	151
9.17	iegenlib::VarTerm Class Reference	151
9.17.1	Detailed Description	154
9.17.2	Constructor & Destructor Documentation	154
9.17.2.1	VarTerm	154
9.17.2.2	VarTerm	155
9.17.2.3	VarTerm	155
9.17.3	Member Function Documentation	156
9.17.3.1	clone	156
9.17.3.2	factorMatches	156
9.17.3.3	isConst	157
9.17.3.4	operator<	157
9.17.3.5	operator=	157

---

9.17.3.6	prettyPrintString	158
9.17.3.7	toString	158
9.17.3.8	type	159
9.17.4	Member Data Documentation	159
9.17.4.1	mSymbol	159
9.18	yy_buffer_state Struct Reference	159
9.18.1	Member Data Documentation	160
9.18.1.1	yy_at_bol	160
9.18.1.2	yy_bs_column	160
9.18.1.3	yy_bs_lineno	160
9.18.1.4	yy_buf_pos	160
9.18.1.5	yy_buf_size	160
9.18.1.6	yy_buffer_status	160
9.18.1.7	yy_ch_buf	160
9.18.1.8	yy_fill_buffer	160
9.18.1.9	yy_input_file	160
9.18.1.10	yy_is_interactive	160
9.18.1.11	yy_is_our_buffer	160
9.18.1.12	yy_n_chars	160
9.19	yy_trans_info Struct Reference	160
9.19.1	Member Data Documentation	161
9.19.1.1	yy_nxt	161
9.19.1.2	yy_verify	161
9.20	yyalloc Union Reference	162
9.20.1	Member Data Documentation	163
9.20.1.1	yyss_alloc	163
9.20.1.2	yyvs_alloc	163
9.21	YYSTYPE Union Reference	164
9.21.1	Member Data Documentation	165
9.21.1.1	conj	165
9.21.1.2	conjlist	165

---

9.21.1.3	elist	165
9.21.1.4	env	165
9.21.1.5	existslist	165
9.21.1.6	exp	165
9.21.1.7	explist	165
9.21.1.8	ival	165
9.21.1.9	relation	165
9.21.1.10	set	165
9.21.1.11	sval	165
9.21.1.12	symlist	165
9.21.1.13	telem	165
<b>10</b>	<b>File Documentation</b>	<b>167</b>
10.1	src/iegenlib/exceptions_test.cc File Reference	167
10.1.1	Detailed Description	168
10.1.2	Function Documentation	169
10.1.2.1	TEST_F	169
10.1.2.2	TEST_F	169
10.1.2.3	TEST_F	170
10.1.2.4	TEST_F	170
10.1.2.5	TEST_F	170
10.1.2.6	TEST_F	171
10.1.2.7	TEST_F	172
10.1.2.8	TEST_F	172
10.1.2.9	TEST_F	172
10.1.2.10	TEST_F	173
10.1.2.11	TEST_F	173
10.2	src/iegenlib/iegenlib.h File Reference	174
10.2.1	Detailed Description	175
10.3	src/iegenlib/parser/gen_parser.cc File Reference	176
10.3.1	Define Documentation	180



---

10.3.1.1 YY_ . . . . .	180
10.3.1.2 YY_LOCATION_PRINT . . . . .	180
10.3.1.3 YY_REDUCE_PRINT . . . . .	180
10.3.1.4 YY_STACK_PRINT . . . . .	180
10.3.1.5 YY_SYMBOL_PRINT . . . . .	180
10.3.1.6 YYABORT . . . . .	180
10.3.1.7 YYACCEPT . . . . .	180
10.3.1.8 YYBACKUP . . . . .	180
10.3.1.9 YYBISON . . . . .	181
10.3.1.10 YYBISON_VERSION . . . . .	181
10.3.1.11 YYCASE_ . . . . .	181
10.3.1.12 yyclearin . . . . .	181
10.3.1.13 YYCOPY . . . . .	181
10.3.1.14 YYCOPY_NEEDED . . . . .	181
10.3.1.15 YYDEBUG . . . . .	181
10.3.1.16 YYDPRINTF . . . . .	181
10.3.1.17 YYEMPTY . . . . .	182
10.3.1.18 YYEOF . . . . .	182
10.3.1.19 YYERRCODE . . . . .	182
10.3.1.20 yyerror . . . . .	182
10.3.1.21 YYERROR . . . . .	182
10.3.1.22 YYERROR_VERBOSE . . . . .	182
10.3.1.23 YYERROR_VERBOSE . . . . .	182
10.3.1.24 YYFAIL . . . . .	182
10.3.1.25 YYFINAL . . . . .	182
10.3.1.26 YYFPRINTF . . . . .	182
10.3.1.27 YYFREE . . . . .	182
10.3.1.28 YYID . . . . .	182
10.3.1.29 YYINITDEPTH . . . . .	182
10.3.1.30 YYLAST . . . . .	182
10.3.1.31 YYLEX . . . . .	182

---

10.3.1.32 YYLLOC_DEFAULT . . . . .	182
10.3.1.33 YYLSP_NEEDED . . . . .	183
10.3.1.34 YYMALLOC . . . . .	183
10.3.1.35 YYMAXDEPTH . . . . .	183
10.3.1.36 YYMAXUTOK . . . . .	183
10.3.1.37 YYNNTS . . . . .	183
10.3.1.38 YYNRULES . . . . .	183
10.3.1.39 YYNSTATES . . . . .	183
10.3.1.40 YYNTOKENS . . . . .	183
10.3.1.41 YYPACT_NINF . . . . .	183
10.3.1.42 yypact_value_is_default . . . . .	183
10.3.1.43 YYPOPSTACK . . . . .	183
10.3.1.44 YYPULL . . . . .	183
10.3.1.45 YYPURE . . . . .	183
10.3.1.46 YYPUSH . . . . .	183
10.3.1.47 YYRECOVERING . . . . .	183
10.3.1.48 YYRHSLOC . . . . .	183
10.3.1.49 YYSIZE_MAXIMUM . . . . .	183
10.3.1.50 YYSIZE_T . . . . .	184
10.3.1.51 YYSKELETON_NAME . . . . .	184
10.3.1.52 YYSTACK_ALLOC . . . . .	184
10.3.1.53 YYSTACK_ALLOC_MAXIMUM . . . . .	184
10.3.1.54 YYSTACK_BYTES . . . . .	184
10.3.1.55 YYSTACK_FREE . . . . .	184
10.3.1.56 YYSTACK_GAP_MAXIMUM . . . . .	184
10.3.1.57 YYSTACK_RELOCATE . . . . .	184
10.3.1.58 yystype . . . . .	184
10.3.1.59 YYSTYPE_IS_DECLARED . . . . .	184
10.3.1.60 YYSTYPE_IS_TRIVIAL . . . . .	184
10.3.1.61 YYSYNTAX_ERROR . . . . .	184
10.3.1.62 YYTABLE_NINF . . . . .	185

---

10.3.1.63	<a href="#">yytable_value_is_error</a>	185
10.3.1.64	<a href="#">YYTERROR</a>	185
10.3.1.65	<a href="#">YYTOKEN_TABLE</a>	185
10.3.1.66	<a href="#">YYTOKENTYPE</a>	185
10.3.1.67	<a href="#">YYTRANSLATE</a>	185
10.3.1.68	<a href="#">YYUNDEFTOK</a>	185
10.3.1.69	<a href="#">YYUSE</a>	185
10.3.2	<a href="#">Typedef Documentation</a>	185
10.3.2.1	<a href="#">YYSTYPE</a>	185
10.3.2.2	<a href="#">yytype_int16</a>	185
10.3.2.3	<a href="#">yytype_int8</a>	185
10.3.2.4	<a href="#">yytype_uint16</a>	185
10.3.2.5	<a href="#">yytype_uint8</a>	185
10.3.3	<a href="#">Enumeration Type Documentation</a>	185
10.3.3.1	<a href="#">yytokentype</a>	185
10.3.4	<a href="#">Function Documentation</a>	186
10.3.4.1	<a href="#">yy_symbol_value_print</a>	186
10.3.4.2	<a href="#">yyerror</a>	186
10.3.4.3	<a href="#">yylex</a>	187
10.3.4.4	<a href="#">yyparse</a>	188
10.3.4.5	<a href="#">yystrlen</a>	188
10.3.5	<a href="#">Variable Documentation</a>	188
10.3.5.1	<a href="#">yychar</a>	189
10.3.5.2	<a href="#">yycheck</a>	189
10.3.5.3	<a href="#">yydefact</a>	189
10.3.5.4	<a href="#">yydefgoto</a>	189
10.3.5.5	<a href="#">yylineno</a>	190
10.3.5.6	<a href="#">yylval</a>	190
10.3.5.7	<a href="#">yyerrs</a>	190
10.3.5.8	<a href="#">yypact</a>	190
10.3.5.9	<a href="#">yypgoto</a>	190

---

10.3.5.10	yyprhs	190
10.3.5.11	yyr1	191
10.3.5.12	yyr2	191
10.3.5.13	yyrhs	191
10.3.5.14	yyrline	192
10.3.5.15	yyrule	192
10.3.5.16	yysrc	192
10.3.5.17	yystos	192
10.3.5.18	yytable	193
10.3.5.19	yytext	193
10.3.5.20	yytname	193
10.3.5.21	yytop	194
10.3.5.22	yytranslate	194
10.3.5.23	yytype	194
10.3.5.24	yyvaluep	194
10.4	src/iegenlib/parser/gen_scanner.cc File Reference	194
10.4.1	Define Documentation	198
10.4.1.1	BEGIN	198
10.4.1.2	ECHO	198
10.4.1.3	EOB_ACT_CONTINUE_SCAN	198
10.4.1.4	EOB_ACT_END_OF_FILE	198
10.4.1.5	EOB_ACT_LAST_MATCH	198
10.4.1.6	FLEX_BETA	198
10.4.1.7	FLEX_SCANNER	198
10.4.1.8	FLEXINT_H	199
10.4.1.9	INITIAL	199
10.4.1.10	INT16_MAX	199
10.4.1.11	INT16_MIN	199
10.4.1.12	INT32_MAX	199
10.4.1.13	INT32_MIN	199
10.4.1.14	INT8_MAX	199

---

10.4.1.15 INT8_MIN . . . . .	199
10.4.1.16 REJECT . . . . .	199
10.4.1.17 UINT16_MAX . . . . .	199
10.4.1.18 UINT32_MAX . . . . .	199
10.4.1.19 UINT8_MAX . . . . .	199
10.4.1.20 unput . . . . .	199
10.4.1.21 YY_AT_BOL . . . . .	199
10.4.1.22 YY_BREAK . . . . .	199
10.4.1.23 YY_BUF_SIZE . . . . .	199
10.4.1.24 YY_BUFFER_EOF_PENDING . . . . .	199
10.4.1.25 YY_BUFFER_NEW . . . . .	199
10.4.1.26 YY_BUFFER_NORMAL . . . . .	199
10.4.1.27 YY_CURRENT_BUFFER . . . . .	199
10.4.1.28 YY_CURRENT_BUFFER_LVALUE . . . . .	200
10.4.1.29 YY_DECL . . . . .	200
10.4.1.30 YY_DECL_IS_OURS . . . . .	200
10.4.1.31 YY_DO_BEFORE_ACTION . . . . .	200
10.4.1.32 YY_END_OF_BUFFER . . . . .	200
10.4.1.33 YY_END_OF_BUFFER_CHAR . . . . .	200
10.4.1.34 YY_EXIT_FAILURE . . . . .	200
10.4.1.35 YY_EXTRA_TYPE . . . . .	200
10.4.1.36 YY_FATAL_ERROR . . . . .	200
10.4.1.37 YY_FLEX_MAJOR_VERSION . . . . .	200
10.4.1.38 YY_FLEX_MINOR_VERSION . . . . .	200
10.4.1.39 YY_FLEX_SUBMINOR_VERSION . . . . .	200
10.4.1.40 YY_FLUSH_BUFFER . . . . .	200
10.4.1.41 YY_INPUT . . . . .	200
10.4.1.42 YY_INT_ALIGNED . . . . .	201
10.4.1.43 YY_LESS_LINENO . . . . .	201
10.4.1.44 YY_MORE_ADJ . . . . .	201
10.4.1.45 yy_new_buffer . . . . .	201

---

10.4.1.46	YY_NEW_FILE	201
10.4.1.47	YY_NO_INPUT	201
10.4.1.48	YY_NULL	201
10.4.1.49	YY_NUM_RULES	201
10.4.1.50	YY_READ_BUF_SIZE	201
10.4.1.51	YY_RESTORE_YY_MORE_OFFSET	201
10.4.1.52	YY_RULE_SETUP	201
10.4.1.53	YY_SC_TO_UI	201
10.4.1.54	yy_set_bol	201
10.4.1.55	yy_set_interactive	201
10.4.1.56	YY_START	202
10.4.1.57	YY_START_STACK_INCR	202
10.4.1.58	YY_STATE_BUF_SIZE	202
10.4.1.59	YY_STATE_EOF	202
10.4.1.60	YY_STRUCT_YY_BUFFER_STATE	202
10.4.1.61	YY_TYPEDEF_YY_BUFFER_STATE	202
10.4.1.62	YY_TYPEDEF_YY_SIZE_T	202
10.4.1.63	YY_USER_ACTION	202
10.4.1.64	yyconst	202
10.4.1.65	yyless	202
10.4.1.66	yyless	202
10.4.1.67	yymore	203
10.4.1.68	YYSTATE	203
10.4.1.69	YYTABLES_NAME	203
10.4.1.70	yyterminate	203
10.4.1.71	yytext_ptr	203
10.4.2	Typedef Documentation	203
10.4.2.1	flex_int16_t	203
10.4.2.2	flex_int32_t	203
10.4.2.3	flex_int8_t	203
10.4.2.4	flex_uint16_t	203

---

10.4.2.5	<code>flex_uint32_t</code>	203
10.4.2.6	<code>flex_uint8_t</code>	203
10.4.2.7	<code>YY_BUFFER_STATE</code>	203
10.4.2.8	<code>YY_CHAR</code>	203
10.4.2.9	<code>yy_size_t</code>	203
10.4.2.10	<code>yy_state_type</code>	203
10.4.3	Function Documentation	203
10.4.3.1	<code>yy_create_buffer</code>	203
10.4.3.2	<code>yy_delete_buffer</code>	204
10.4.3.3	<code>yy_fatal_error</code>	204
10.4.3.4	<code>yy_fatal_error</code>	204
10.4.3.5	<code>yy_flush_buffer</code>	204
10.4.3.6	<code>yy_get_next_buffer</code>	205
10.4.3.7	<code>yy_get_previous_state</code>	205
10.4.3.8	<code>yy_init_buffer</code>	205
10.4.3.9	<code>yy_init_globals</code>	206
10.4.3.10	<code>yy_load_buffer_state</code>	206
10.4.3.11	<code>yy_scan_buffer</code>	206
10.4.3.12	<code>yy_scan_bytes</code>	207
10.4.3.13	<code>yy_scan_string</code>	208
10.4.3.14	<code>yy_switch_to_buffer</code>	209
10.4.3.15	<code>yy_try_NUL_trans</code>	209
10.4.3.16	<code>yyalloc</code>	210
10.4.3.17	<code>yyensure_buffer_stack</code>	210
10.4.3.18	<code>yyfree</code>	211
10.4.3.19	<code>yyget_debug</code>	211
10.4.3.20	<code>yyget_extra</code>	211
10.4.3.21	<code>yyget_in</code>	211
10.4.3.22	<code>yyget_leng</code>	211
10.4.3.23	<code>yyget_lineno</code>	211
10.4.3.24	<code>yyget_out</code>	211

---

10.4.3.25	<a href="#">yyget_text</a>	211
10.4.3.26	<a href="#">yylex</a>	212
10.4.3.27	<a href="#">yylex_destroy</a>	212
10.4.3.28	<a href="#">yypop_buffer_state</a>	212
10.4.3.29	<a href="#">yypush_buffer_state</a>	213
10.4.3.30	<a href="#">yyrealloc</a>	214
10.4.3.31	<a href="#">yyrestart</a>	214
10.4.3.32	<a href="#">yyset_debug</a>	214
10.4.3.33	<a href="#">yyset_extra</a>	214
10.4.3.34	<a href="#">yyset_in</a>	214
10.4.3.35	<a href="#">yyset_lineno</a>	214
10.4.3.36	<a href="#">yyset_out</a>	214
10.4.3.37	<a href="#">yywrap</a>	215
10.4.4	Variable Documentation	215
10.4.4.1	<a href="#">yy_accept</a>	215
10.4.4.2	<a href="#">yy_base</a>	215
10.4.4.3	<a href="#">yy_buffer_stack</a>	215
10.4.4.4	<a href="#">yy_buffer_stack_max</a>	215
10.4.4.5	<a href="#">yy_buffer_stack_top</a>	216
10.4.4.6	<a href="#">yy_c_buf_p</a>	216
10.4.4.7	<a href="#">yy_chk</a>	216
10.4.4.8	<a href="#">yy_def</a>	216
10.4.4.9	<a href="#">yy_did_buffer_switch_on_eof</a>	216
10.4.4.10	<a href="#">yy_ec</a>	216
10.4.4.11	<a href="#">yy_flex_debug</a>	216
10.4.4.12	<a href="#">yy_hold_char</a>	216
10.4.4.13	<a href="#">yy_init</a>	216
10.4.4.14	<a href="#">yy_last_accepting_cpos</a>	216
10.4.4.15	<a href="#">yy_last_accepting_state</a>	216
10.4.4.16	<a href="#">yy_meta</a>	216
10.4.4.17	<a href="#">yy_n_chars</a>	217



---

10.4.4.18	yy_nxt	217
10.4.4.19	yy_start	217
10.4.4.20	yyin	217
10.4.4.21	yylen	217
10.4.4.22	yylineno	217
10.4.4.23	yyout	217
10.4.4.24	yytext	217
10.5	src/iegenlib/parser/parser.cc File Reference	217
10.5.1	Detailed Description	218
10.5.2	Function Documentation	219
10.5.2.1	yparse	219
10.6	src/iegenlib/parser/parser.h File Reference	220
10.6.1	Detailed Description	221
10.7	src/iegenlib/parser/parser_test.cc File Reference	222
10.7.1	Detailed Description	224
10.7.2	Function Documentation	225
10.7.2.1	TEST	225
10.7.2.2	TEST	225
10.7.2.3	TEST	226
10.7.2.4	TEST	226
10.7.2.5	TEST	227
10.7.2.6	TEST	227
10.7.2.7	TEST	228
10.7.2.8	TEST	228
10.7.2.9	TEST	229
10.7.2.10	TEST	229
10.7.2.11	TEST	230
10.7.2.12	TEST	230
10.7.2.13	TEST	231
10.7.2.14	TEST	231
10.7.2.15	TEST	232

---

10.7.2.16 TEST . . . . .	232
10.7.2.17 TEST . . . . .	233
10.7.2.18 TEST . . . . .	233
10.7.2.19 TEST . . . . .	234
10.7.2.20 TEST . . . . .	234
10.7.2.21 TEST . . . . .	235
10.7.2.22 TEST . . . . .	235
10.7.2.23 TEST . . . . .	236
10.7.2.24 TEST . . . . .	236
10.7.2.25 TEST . . . . .	237
10.7.2.26 TEST . . . . .	237
10.7.2.27 TEST . . . . .	238
10.7.2.28 TEST . . . . .	238
10.7.2.29 TEST . . . . .	239
10.7.2.30 TEST . . . . .	239
10.7.2.31 TEST . . . . .	240
10.7.2.32 TEST . . . . .	240
10.7.2.33 TEST . . . . .	241
10.7.2.34 TEST . . . . .	241
10.7.2.35 TEST . . . . .	242
10.7.2.36 TEST . . . . .	242
10.7.2.37 TEST . . . . .	243
10.7.2.38 TEST . . . . .	243
10.7.2.39 TEST . . . . .	244
10.7.2.40 TEST . . . . .	244
10.7.2.41 TEST . . . . .	245
10.7.2.42 TEST . . . . .	245
10.7.2.43 TEST . . . . .	246
10.7.2.44 TEST . . . . .	246
10.7.2.45 TEST . . . . .	246
10.7.2.46 TEST . . . . .	247

---

10.7.2.47 TEST . . . . .	247
10.7.2.48 TEST . . . . .	248
10.7.2.49 TEST . . . . .	248
10.7.2.50 TEST . . . . .	249
10.7.2.51 TEST . . . . .	249
10.7.2.52 TEST . . . . .	250
10.7.2.53 TEST . . . . .	250
10.7.2.54 TEST . . . . .	251
10.7.2.55 TEST . . . . .	251
10.8 src/iegenlib/set_relation/environment.cc File Reference . . . . .	251
10.8.1 Detailed Description . . . . .	253
10.9 src/iegenlib/set_relation/environment.h File Reference . . . . .	253
10.9.1 Detailed Description . . . . .	255
10.10src/iegenlib/set_relation/environment_test.cc File Reference . . . . .	256
10.10.1 Detailed Description . . . . .	256
10.10.2 Function Documentation . . . . .	257
10.10.2.1 TEST_F . . . . .	257
10.10.2.2 TEST_F . . . . .	258
10.10.2.3 TEST_F . . . . .	258
10.10.2.4 TEST_F . . . . .	259
10.10.2.5 TEST_F . . . . .	259
10.11src/iegenlib/set_relation/expression.cc File Reference . . . . .	259
10.11.1 Detailed Description . . . . .	260
10.12src/iegenlib/set_relation/expression.h File Reference . . . . .	261
10.12.1 Detailed Description . . . . .	262
10.13src/iegenlib/set_relation/expression_test.cc File Reference . . . . .	263
10.13.1 Detailed Description . . . . .	264
10.13.2 Function Documentation . . . . .	265
10.13.2.1 TEST_F . . . . .	265
10.13.2.2 TEST_F . . . . .	266
10.13.2.3 TEST_F . . . . .	266

---

10.13.2.4 TEST_F	267
10.13.2.5 TEST_F	267
10.13.2.6 TEST_F	268
10.13.2.7 TEST_F	268
10.13.2.8 TEST_F	269
10.13.2.9 TEST_F	269
10.13.2.10 TEST_F	270
10.13.2.11 TEST_F	270
10.13.2.12 TEST_F	270
10.13.2.13 TEST_F	271
10.13.2.14 TEST_F	271
10.13.2.15 TEST_F	271
10.14src/iegenlib/set_relation/set_relation.cc File Reference	272
10.14.1 Detailed Description	272
10.15src/iegenlib/set_relation/set_relation.h File Reference	273
10.15.1 Detailed Description	274
10.16src/iegenlib/set_relation/set_relation_test.cc File Reference	275
10.16.1 Detailed Description	277
10.16.2 Function Documentation	277
10.16.2.1 TEST_F	277
10.16.2.2 TEST_F	278
10.16.2.3 TEST_F	278
10.16.2.4 TEST_F	279
10.16.2.5 TEST_F	279
10.16.2.6 TEST_F	280
10.16.2.7 TEST_F	280
10.16.2.8 TEST_F	281
10.16.2.9 TEST_F	281
10.16.2.10 TEST_F	282
10.16.2.11 TEST_F	282
10.16.2.12 TEST_F	283

---

10.16.2.13TEST_F	283
10.16.2.14TEST_F	284
10.16.2.15TEST_F	284
10.16.2.16TEST_F	285
10.16.2.17TEST_F	285
10.16.2.18TEST_F	286
10.16.2.19TEST_F	286
10.16.2.20TEST_F	287
10.16.2.21TEST_F	287
10.16.2.22TEST_F	288
10.16.2.23TEST_F	288
10.16.2.24TEST_F	288
10.16.2.25TEST_F	289
10.16.2.26TEST_F	289
10.16.2.27TEST_F	290
10.16.2.28TEST_F	290
10.16.2.29TEST_F	291
10.16.2.30TEST_F	291
10.16.2.31TEST_F	292
10.16.2.32TEST_F	292
10.16.2.33TEST_F	293
10.16.2.34TEST_F	293
10.16.2.35TEST_F	294
10.16.2.36TEST_F	294
10.16.2.37TEST_F	295
10.16.2.38TEST_F	295
10.16.2.39TEST_F	296
10.16.2.40TEST_F	296
10.16.2.41TEST_F	297
10.16.2.42TEST_F	297
10.16.2.43TEST_F	298

---

10.16.2.44	TEST_F	298
10.16.2.45	TEST_F	299
10.16.2.46	TEST_F	299
10.16.2.47	TEST_F	300
10.16.2.48	TEST_F	300
10.17	src/iegenlib_calc.cc File Reference	301
10.17.1	Define Documentation	302
10.17.1.1	AST_DOT_FILE_NAME	302
10.17.1.2	SET_RELATION_DOT_FILE_NAME	302
10.17.2	Function Documentation	302
10.17.2.1	contains	302
10.17.2.2	handleApply	303
10.17.2.3	handleCompose	304
10.17.2.4	handleInverse	305
10.17.2.5	handleRelation	306
10.17.2.6	handleSet	306
10.17.2.7	handleUnion	307
10.17.2.8	isRelation	308
10.17.2.9	lowercase	309
10.17.2.10	main	310
10.17.2.11	newRelationOrNull	310
10.17.2.12	newSetOrNull	311

# Chapter 1

## IEGen C++ Rewrite

### Authors

Michelle Strout  
Joe Strout  
Alan LaMielle  
Catherine Olschanowsky  
Barbara Kreaseck  
Mark Heim  
Ian Craig  
Nicholas Jeanette

### Date

Date Started: 5/17/10

### 1.1 Intro

IEGenLib is a library with data structures and routines that can represent, parse, and visit integer tuple sets and relations with affine constraints and uninterpreted function symbol equality constraints. See `src/parser/parser_test.cc` for examples of strings that can be parsed.

The `run_tests.cc` driver will execute all tests identified with the `TEST` macro. In each of the `src/` subdirectories, the `*_test.cc` files define regression tests.

The `iegenlib_calc` binary will execute tests that you input yourself and will return the interpreted version of the input along with a dot file to the file `set_relation.dot` in the directory where the program is executed.

## 1.2 COPYING

Copyright (c) 2009,2010,2011,2012 Colorado State University  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 1.3 README

```
=====
README for IGenLib
=====
IGenLib is a library that represents and manipulates integer sets and
relations that have affine and uninterpreted function constraints.

==== Unpacking the Distribution =====

The distribution is shipped as a tar file. When you unpack the tar
file it creates a directory that is the root of the distribution.

    tar xzvf iegenlib-1.0.0.tgz

==== Quick Start =====

    A. Running the IGenLib Calculator
    B. Using the IGenLib Python Interface

This distribution is shipped with LINUX-64 bit and Mac OSX binaries in
addition to the sources and build system needed to compile and link on
unix-based machines.

In the below, assume the IEGEN_HOME environment variable is the path
to the top installation directory. Two installation directories come
with the package:
```



```
iegen-MACOSX/
and
iegen-LINUX64/.
```

---- A. Running the IEGenLib calculator -----

The `iegenlib_calc` binary can be executed to enter sets and relations and operations on them. When run, the `iegenlib_calc` suggests examples.

```
$IEGEN_HOME/bin/iegenlib_calc
```

The program will present a prompt for an input string. You may input any valid string and it will return the interpreted string and a formatted dot file to a file called `set_relation.dot`. The calculator continues until a blank line is input or [CTRL][D] is entered.

---- B. Using the IEGenLib Python bindings -----

Required to use Python bindings to IEGenLib:  
\* Python 2.7.2 known to work on linux and Mac

The Python bindings shipped in the distribution (`iegen/bindings/..`) can also be used to interface to the library.

(1) Set the PYTHONPATH environment variable

```
export PYTHONPATH=$PYTHONPATH:$IEGEN_HOME/bindings
- OR -
setenv PYTHONPATH $IEGEN_HOME/bindings
```

(2) Run python

```
python
>>> import iegenlib
```

(3) Examples

The following Python statements are examples on how to create sets and relations:

```
>>> S1 = iegenlib.Set("[s,i]: 0<=s && s<T && 0<=i && i<N")
>>> print S1
```

```
>>> R1 = iegenlib.Relation("[s,i]->[0,s,1,i,0]")
```

NOTE: Both the omega and ISL syntax are allowed for specifying sets and relations. Some of the short cuts such as `0<=i<N` are missing however.

The following Python statements are examples on how to apply operations (such as Apply, Union, Inverse, and Compose) between sets/relations:

```
>>> S2 = R1.Apply(S1)           # Apply operation #
>>> S3 = iegenlib.Set("[i,j]: 0<=i and i<n and 5<=j and j<m)")
```

```

>>> S4 = iegenlib.Set("{[i,j]: 0<=i and i<n and m+5<=j and j<m+10}")
>>> S4 = S3.Union(S4)           # Union operation #
>>> R2 = iegenlib.Relation("{[i,j]->[ip,jp]: ip=f(i) and jp=2j}")
>>> R3 = iegenlib.Relation("{[q,r]->[i,j]: i=q and j=r}")
>>> R3 = R3.Compose(R2)        # Compose operation#
>>> R2 = R2.Inverse()         # Inverse operation#
>>> print R2                   # to print the resulting set/relation#
    { [ip, jp] -> [i, j] : ip - f(i) = 0 && 2 j - jp = 0 }
    # The output of the print command #

```

Some examples that use function inverse declarations.

```

>>> iegenlib.appendCurrEnv("f() = inverse f_inv()");
>>> S5 = iegenlib.Set("{[i,j]:i=f(f_inv(j))}")
>>> print "S5 = ", S5
>>> R4 = iegenlib.Relation("{[i,j]->[ip,j]: ip=f(i)}")
>>> print "R4 = ", R4
>>> S6 = R4.Apply(S5)
>>> print "S6 = ", S6

```

The user can obtain dot files for SparseConstraints objects of sets/relations from the IEGenlib Python Interface by using `toDotString()` methods. Python methods `open(filename, mode)` and `write(item)` can be used to open and write to the output file, as in the following examples:

```

>>> S3 = iegenlib.Set("[n,m]->{[i,j]: 0<=i and i<n and 5<=j and j<m}")
>>> file = open("S3.dot", 'w')
>>> file.write(S3.toDotString())
>>> file.close()

```

#### ==== Distribution Organization =====

The distribution is shipped as a tar file. When you unpack the tar file it creates a directory that is the root of the distribution.

```
tar xzvf iegenlib-#.##.tgz
```

This distribution is shipped with LINUX-64 bit (`iegen-LINUX64/`) and Mac OSX (`iegen-MACOSX/`) binaries in addition to the sources and build system needed to compile and link on unix-based machines.

```
iegen-LINUX64/ and iegen-MACOSX/
  lib/
    libiegenlib.a          // C++ library

  bin/
    iegenlib_calc         // interactive calculator
    run_iegenlib_tests    // regression tests

  bindings/               // Python bindings
    _iegenlib.so
    iegenlib.py
```

Generated documentation can be found in the doc/refman.pdf file.

The refman.pdf file and a full set of html documentation can also be found at the project downloads page website.

Sources for the project are found in the distribution root src/sub-directory.

The library and demonstration driver create both string and dot output; graphviz is needed to visualize the dot output.

==== Running the regression tests =====

(1) Set the IEGEN\_HOME environment variable :

```
On a Mac, from the distribution root directory:
  export IEGEN_HOME=`pwd`/iegen-MACOSX
  - OR -
  setenv IEGEN_HOME `pwd`/iegen-MACOSX
```

```
On a Linux, from the distribution root directory:
  export IEGEN_HOME=`pwd`/iegen-LINUX64
  - OR -
  setenv IEGEN_HOME `pwd`/iegen-LINUX64
```

(2) run the regression tests

```
$IEGEN_HOME/bin/run_iegenlib_tests
```

==== Building IEGenLib from Source =====

Build Command Sequence

```
./configure
make install
```

Build and Test without creating an install directory

```
./configure
make test
```

Required to use Python bindings to IEGenLib:

\* Python 2.7.2 known to work on linux and Mac

## Requirements

- \* cmake 2.6 or newer
  
- \* C++ compiler, the below versions are known to work
  - Mac i686-apple-darwin10-g++-4.2.1,
  - GCC 4.6.3 20120306 (Red Hat 4.6.3-2)
  
- \* If re-generating parser files BISON 2.4 and FLEX 2.5 or newer
  
- \* If re-generating docs
  - \* doxygen (<http://www.stack.nl/~dimitri/doxygen/index.html>, known to work with 1.5.6 and 1.7.5)
  - \* dot (<http://www.graphviz.org>)
  
- \* If rebuilding python bindings from source:
  - \* swig version 2.0.4 for linux, version 1.3.40 for Mac known to work
  - \*\* please note: if installing swig from macports 2 packages are required: swig and swig-python
  
- \* If you plan to change the grammar for the sets or relations:
  - \* Flex: version 2.5.35 known to work
  - \* Bison: version 2.4.1 known to work

Please notify us if you find that other versions of these tools that work or do not work for you.

Build files for the project are generated using cmake (except for the distribution root-level Makefile that is included in the tar file). Cmake version 2.6 or newer is required. Make files can be generated to only build the binaries, to additionally generate the parser files, and/or to additionally include Python bindings to the library. These options are enabled through a configure script in the root of the project. If the parser option is included, then additionally FLEX version 2.5 or newer and BISON version 2.4 or newer are required. If documentation is to be re-generated, then doxygen and dot are also required to create it.

Run `./configure --help` for more build configuration options.

The default install directory is `${IEGEN_DIR}/iegen/`. This directory may be changed using options to the configure script. Note that the cmake system places binaries in an `iegen/` directory, rather than an `iegen-LINUX64/` or `iegen-MacOSX/` directory as are shipped in the distribution.

==== Running Tests ====

The `iegen/bin/run_iegenlib_tests` binary can be executed to run the tests of the various components of IEGenLib. Additionally, from the root of the distribution you can run `'make test'` to run these same tests.

The gtest framework is used for writing and running unit tests. Information on gtest can be found at:

<http://code.google.com/p/googletest/>

and introductory documentation is at:

[http://code.google.com/p/googletest/wiki/V1\\_6\\_Primer](http://code.google.com/p/googletest/wiki/V1_6_Primer)

==== Documentation ====

To create the doxygen documentation, run the following command:

```
make docs
```

An index.html file will be created in the distribution root directory, in doc/html/. A latex pdf file, called refman.pdf will also be created, in the doc/latex/ directory.

==== Debug and Release Builds ====

The default build is the debug build. It is set by cmake in the configure script with `-DCMAKE_BUILD_TYPE:String=Debug`. The compiler options for the debug build are `'-O0 -g'`.

To create a release version, use in the `-build--release` option to the configure script. The compiler options for the release are `'-O3 -DNDEBUG'`.

To show actual build commands, you can run the verbose version of the build:

```
make install VERBOSE=1
```

==== Contact ====

For more information about the PIES project, please visit this URL:

<http://www.cs.colostate.edu/hpc/PIES/>

Or, please contact the PIES project director:

Dr. Michelle Strout, [mstrout@cs.colostate.edu](mailto:mstrout@cs.colostate.edu)



## Chapter 2

# Directory Hierarchy

### 2.1 Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

src	23
iegenlib	20
parser	21
set_relation	22





# Chapter 3

## Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

- [iegenlib](#) . . . . . 25
- [iegenlib::parser](#) . . . . . 29



# Chapter 4

## Class Index

### 4.1 Class Hierarchy

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

iegenlib::Conjunction . . . . .	41
iegenlib::Environment . . . . .	60
EnvironmentTest . . . . .	65
ExceptionTestExp . . . . .	66
ExceptionTestParser . . . . .	66
ExceptionTestSetRelation . . . . .	67
iegenlib::Exp . . . . .	67
ExpTest . . . . .	85
SetRelationTest . . . . .	105
iegenlib::SparseConstraints . . . . .	107
iegenlib::Relation . . . . .	86
iegenlib::Set . . . . .	98
iegenlib::Term . . . . .	115
iegenlib::TupleVarTerm . . . . .	131
iegenlib::UFCallTerm . . . . .	140
iegenlib::VarTerm . . . . .	151
iegenlib::TupleElemDecl . . . . .	128
yy_buffer_state . . . . .	159
yy_trans_info . . . . .	160
yyalloc . . . . .	162
YYSTYPE . . . . .	164



# Chapter 5

## Class Index

### 5.1 Class List

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

<a href="#">iegenlib::Conjunction</a>	
Class containing sets of all the equalities and inequalities . . . . .	41
<a href="#">iegenlib::Environment</a> . . . . .	60
<a href="#">EnvironmentTest</a> . . . . .	65
<a href="#">ExceptionTestExp</a>	
Class to test exception handling in expression/ methods . . . . .	66
<a href="#">ExceptionTestParser</a>	
Class to test exception handling in parser/ methods . . . . .	66
<a href="#">ExceptionTestSetRelation</a>	
Class to test exception handling in set_relation/ methods . . . . .	67
<a href="#">iegenlib::Exp</a>	
An affine expression that allows uninterpreted function call terms . . .	67
<a href="#">ExpTest</a> . . . . .	85
<a href="#">iegenlib::Relation</a>	
A <a href="#">SparseConstraints</a> class that represents a <a href="#">Relation</a> . . . . .	86
<a href="#">iegenlib::Set</a>	
A <a href="#">SparseConstraints</a> class that represents a <a href="#">Set</a> . . . . .	98
<a href="#">SetRelationTest</a> . . . . .	105
<a href="#">iegenlib::SparseConstraints</a>	
Base class that contains the conjunctions and a pointer to an envi- ronment . . . . .	107
<a href="#">iegenlib::Term</a>	
A coefficient multiplied by one. Subclasses are multiplied by other entities . . . . .	115

---

<a href="#">iegenlib::TupleElemDecl</a>	
Info about a tuple element, which is a constant or a variable . . . . .	128
<a href="#">iegenlib::TupleVarTerm</a>	
Represents a coefficient multiplied by a tuple variable . . . . .	131
<a href="#">iegenlib::UFCallTerm</a>	
Represents a coefficient multiplied by an uninterpreted function call	140
<a href="#">iegenlib::VarTerm</a>	
Represents a coefficient multiplied by a variable or symbolic constant	151
<a href="#">yy_buffer_state</a> . . . . .	159
<a href="#">yy_trans_info</a> . . . . .	160
<a href="#">yyalloc</a> . . . . .	162
<a href="#">YYSTYPE</a> . . . . .	164

# Chapter 6

## File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

<a href="#">src/iegenlib_calc.cc</a>	301
<a href="#">src/iegenlib/exceptions_test.cc</a>	
Test exceptions for the parser/, set_relation/ and util/ classes	167
<a href="#">src/iegenlib/iegenlib.h</a>	
Main IEGen header file	174
<a href="#">src/iegenlib/parser/gen_parser.cc</a>	176
<a href="#">src/iegenlib/parser/gen_scanner.cc</a>	194
<a href="#">src/iegenlib/parser/parser.cc</a>	
Implementations to methods declared in <a href="#">parser.h</a>	217
<a href="#">src/iegenlib/parser/parser.h</a>	
Namespace to allow the user to make use of the parser	220
<a href="#">src/iegenlib/parser/parser_test.cc</a>	
Test file for parsing sets and relations from an omega-like syntax	222
<a href="#">src/iegenlib/set_relation/environment.cc</a>	
Implementations of the Environment class	251
<a href="#">src/iegenlib/set_relation/environment.h</a>	
Declarations for the Environment class	253
<a href="#">src/iegenlib/set_relation/environment_test.cc</a>	
Test for the Environment class	256
<a href="#">src/iegenlib/set_relation/expression.cc</a>	
Implementations of the expression classes	259
<a href="#">src/iegenlib/set_relation/expression.h</a>	
Declarations for expression and related classes	261
<a href="#">src/iegenlib/set_relation/expression_test.cc</a>	
Test for the Expression classes and Term classes	263

src/iegenlib/set_relation/ <a href="#">set_relation.cc</a>	
Implementations of the Set and Relation classes . . . . .	272
src/iegenlib/set_relation/ <a href="#">set_relation.h</a>	
Declarations for the set and relation classes and classes they contain except for the Exp and Term classes . . . . .	273
src/iegenlib/set_relation/ <a href="#">set_relation_test.cc</a>	
Set and Relation tests . . . . .	275

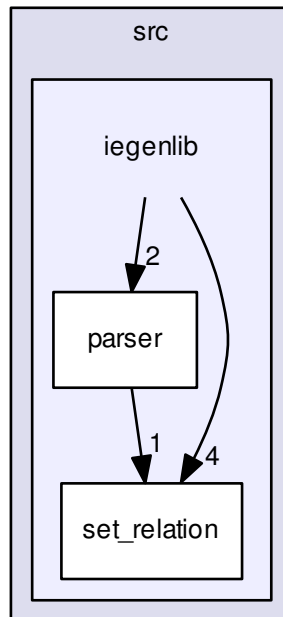


## **Chapter 7**

# **Directory Documentation**

## 7.1 src/iegenlib/ Directory Reference

Directory dependency graph for src/iegenlib/:



### Directories

- directory [parser](#)
- directory [set\\_relation](#)

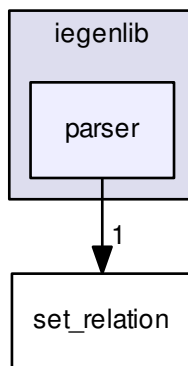
### Files

- file [exceptions\\_test.cc](#)  
*Test exceptions for the parser/, set\_relation/ and util/ classes.*
- file [iegenlib.h](#)

*Main IEGen header file.*

## 7.2 src/iegenlib/parser/ Directory Reference

Directory dependency graph for src/iegenlib/parser/:



### Files

- file [gen\\_parser.cc](#)
- file [gen\\_scanner.cc](#)
- file [parser.cc](#)

*Implementations to methods declared in [parser.h](#).*

- file [parser.h](#)

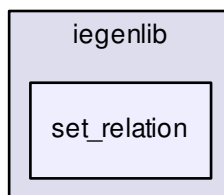
*a namespace to allow the user to make use of the parser*

- file [parser\\_test.cc](#)

*Test file for parsing sets and relations from an omega-like syntax.*

### 7.3 src/iegenlib/set\_relation/ Directory Reference

Directory dependency graph for src/iegenlib/set\_relation/:

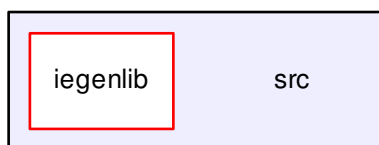


#### Files

- file [environment.cc](#)  
*Implementations of the Environment class.*
- file [environment.h](#)  
*Declarations for the Environment class.*
- file [environment\\_test.cc](#)  
*Test for the Environment class.*
- file [expression.cc](#)  
*Implementations of the expression classes.*
- file [expression.h](#)  
*Declarations for expression and related classes.*
- file [expression\\_test.cc](#)  
*Test for the Expression classes and Term classes.*
- file [set\\_relation.cc](#)  
*Implementations of the Set and Relation classes.*
- file [set\\_relation.h](#)  
*Declarations for the set and relation classes and classes they contain except for the Exp and Term classes.*
- file [set\\_relation\\_test.cc](#)  
*Set and Relation tests.*

## 7.4 src/ Directory Reference

Directory dependency graph for src/:



### Directories

- directory [iegenlib](#)

### Files

- file [iegenlib\\_calc.cc](#)



## Chapter 8

# Namespace Documentation

### 8.1 eigenlib Namespace Reference

#### Namespaces

- namespace [parser](#)

#### Classes

- class [Environment](#)
- class [Term](#)  
*A coefficient multiplied by one. Subclasses are multiplied by other entities.*
- class [UFCallTerm](#)  
*Represents a coefficient multiplied by an uninterpreted function call.*
- class [TupleVarTerm](#)  
*Represents a coefficient multiplied by a tuple variable.*
- class [VarTerm](#)  
*Represents a coefficient multiplied by a variable or symbolic constant.*
- class [Exp](#)  
*An affine expression that allows uninterpreted function call terms.*
- class [TupleElemDecl](#)  
*Info about a tuple element, which is a constant or a variable.*
- class [Conjunction](#)  
*Class containing sets of all the equalities and inequalities.*
- class [SparseConstraints](#)  
*Base class that contains the conjunctions and a pointer to an environment.*

- class [Set](#)

A [SparseConstraints](#) class that represents a [Set](#).

- class [Relation](#)

A [SparseConstraints](#) class that represents a [Relation](#).

## Functions

- void [setCurrEnv](#) ()

*Resets the current environment to empty.*

- void [setCurrEnv](#) (std::string str)

*Sets the global environment after creating one by parsing the string.*

- void [appendCurrEnv](#) (std::string str)

*append to this environment*

- std::string [queryInverseCurrEnv](#) (std::string funcName)

*search this environment for a function inverse*

## Variables

- [Environment currentEnv](#)

### 8.1.1 Detailed Description

The parser namespace hides the set and relation lexer and parser functions and associated global variables.

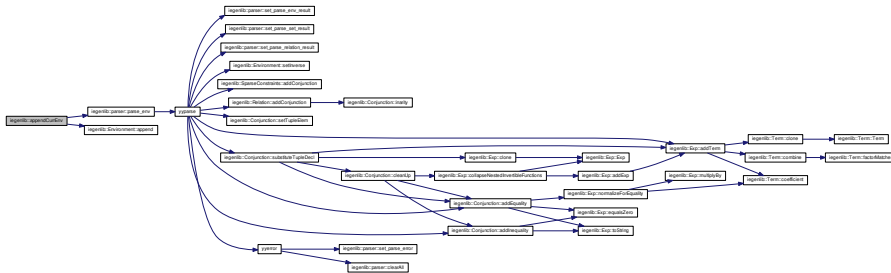
### 8.1.2 Function Documentation

#### 8.1.2.1 void `iegenlib::appendCurrEnv ( std::string str )`

append to this environment



Here is the call graph for this function:



Here is the caller graph for this function:



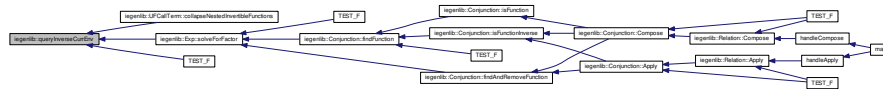
8.1.2.2 std::string iegenlib::queryInverseCurrEnv ( std::string funcName )

search this environment for a function inverse

Here is the call graph for this function:



Here is the caller graph for this function:

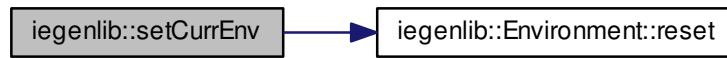


### 8.1.2.3 void iegenlib::setCurrEnv ( )

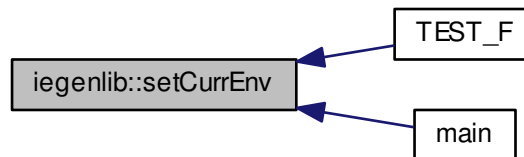
Resets the current environment to empty.

Sets the global environment after creating one.

Here is the call graph for this function:



Here is the caller graph for this function:

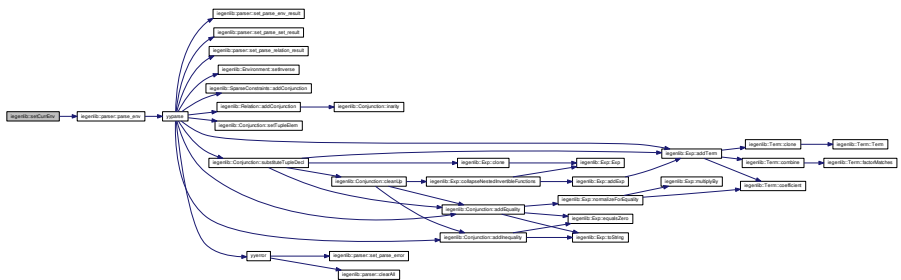


8.1.2.4 void iegenlib::setCurrEnv ( std::string str )

Sets the global environment after creating one by parsing the string.

Resets the current environment to empty and then parses the string to create a new environment. Release 1 must be of the form "f() = inverse f'()" where f and f' can be any identifier.

Here is the call graph for this function:



8.1.3 Variable Documentation

8.1.3.1 Environment iegenlib::currentEnv

8.2 iegenlib::parser Namespace Reference

Functions

- int [string\\_get\\_next\\_char](#) ()
  - Used by lexer to obtain each character of input string being parsed.*
- Environment \* [parse\\_env](#) (std::string env\_string)
- Environment \* [get\\_parse\\_env\\_result](#) ()
- void [set\\_parse\\_env\\_result](#) (Environment \*e)
- Set \* [parse\\_set](#) (string set\_string)
- Set [get\\_parse\\_set\\_result](#) ()
- void [set\\_parse\\_set\\_result](#) (Set \*s)
- Relation \* [parse\\_relation](#) (string relation\_string)
- Relation [get\\_parse\\_relation\\_result](#) ()
- void [set\\_parse\\_relation\\_result](#) (Relation \*s)
- void [set\\_parse\\_error](#) (string error)
- bool [get\\_parse\\_error](#) ()
- void [clearAll](#) ()

- [Set](#) \* [parse\\_set](#) (std::string set\_string)
- [Relation](#) \* [parse\\_relation](#) (std::string relation\_string)
- void [set\\_parse\\_error](#) (std::string err)

## Variables

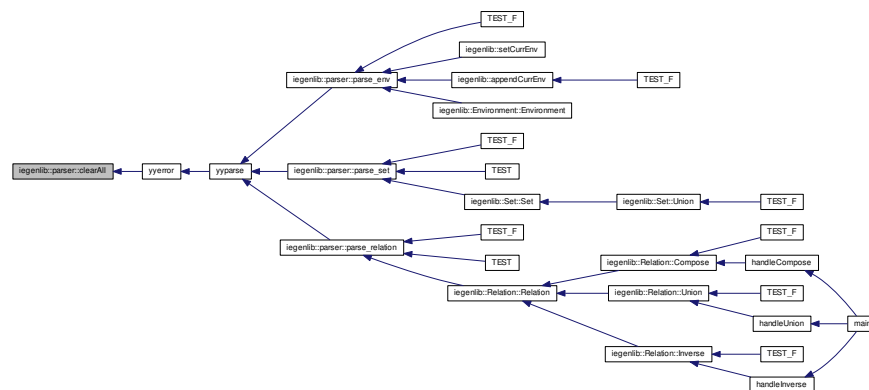
- string [input\\_buffer](#)
- unsigned int [input\\_pos](#)
- bool [parse\\_error](#)
- string [error\\_message](#)  
*error message for when an error occurs in the parser*
- [Environment](#) \* [parse\\_env\\_result](#)
- [Relation](#) \* [parse\\_relation\\_result](#)
- [Set](#) \* [parse\\_set\\_result](#)

## 8.2.1 Function Documentation

### 8.2.1.1 void [iegenlib::parser::clearAll](#) ( )

`clearAll` frees (if needed) and resets the `parse_relation_result` and the `parse_set_result` so no data is left in those variables

Here is the caller graph for this function:



### 8.2.1.2 Environment\* iegenlib::parser::get\_parse\_env\_result ( )

getter for the parse\_env\_result

#### Returns

Environment\* parse\_env\_result

### 8.2.1.3 bool iegenlib::parser::get\_parse\_error ( )

getter for the parse\_error flag

#### Returns

bool parse\_error

### 8.2.1.4 Relation iegenlib::parser::get\_parse\_relation\_result ( )

getter for the parse\_relation\_result the getter can only be called a single time the code that calls it gets a copy of the object and we delete our copy

#### Returns

[Relation](#) parse\_relation\_result

### 8.2.1.5 Set iegenlib::parser::get\_parse\_set\_result ( )

getter for the parse\_set\_result

#### Returns

[Set](#) parse\_set\_result

### 8.2.1.6 Environment \* iegenlib::parser::parse\_env ( std::string env\_string )

parse\_env passes a string representation of an environment to the parser and an [Environment](#) gets created.

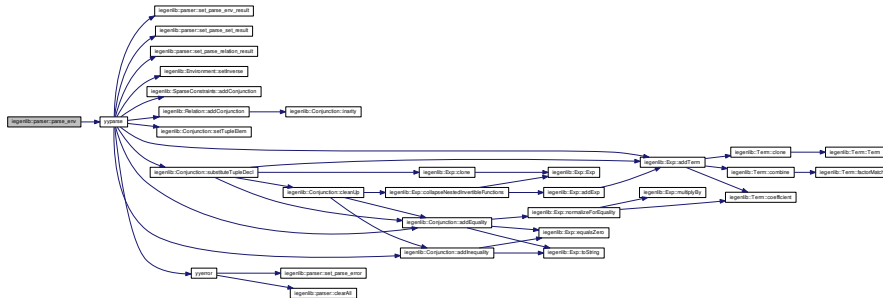
#### Parameters

<i>string</i>	env string that is to be parsed
---------------	---------------------------------

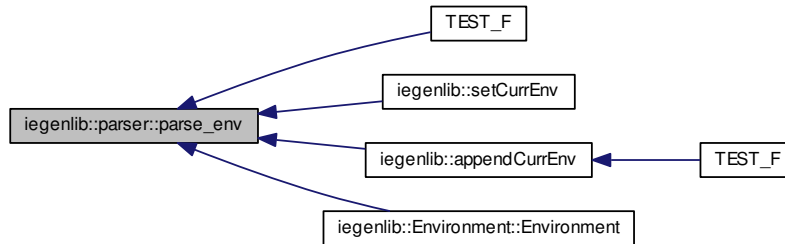
### Returns

Null pointer if error occurs or an environment is returned  
[Environment](#) pointer when parsing is successful

Here is the call graph for this function:



Here is the caller graph for this function:



#### 8.2.1.7 Relation \* iegenlib::parser::parse\_relation ( std::string relation\_string )

passes a relation string into the parser and creates the [Relation](#)

#### Parameters

<i>string</i>	relation string that is to be parsed
---------------	--------------------------------------

Returns

Null pointer if error occurs or a set is returned  
[Relation](#) pointer when parsing is successful

8.2.1.8 Relation\* iegenlib::parser::parse\_relation ( string relation\_string )

passes a relation string into the parser and creates the [Relation](#)

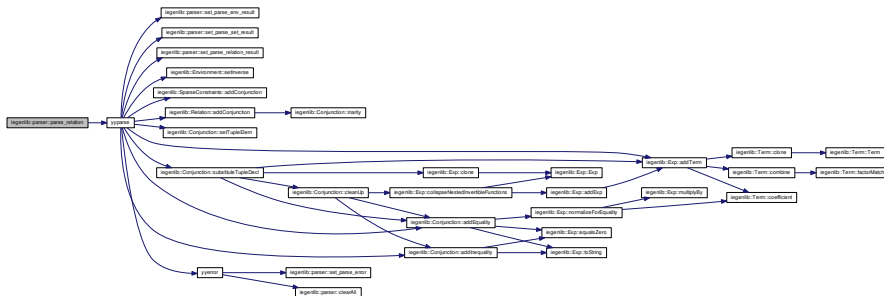
Parameters

<i>string</i>	relation string that is to be parsed
---------------	--------------------------------------

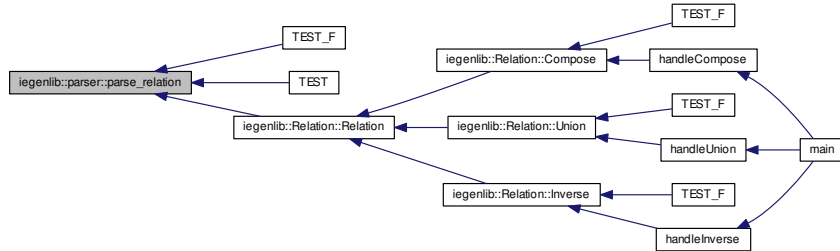
Returns

Null pointer if error occurs or a set is returned  
[Relation](#) pointer when parsing is successful

Here is the call graph for this function:



Here is the caller graph for this function:



#### 8.2.1.9 Set \* iegenlib::parser::parse\_set ( std::string set\_string )

parse\_set passes a string representation of a set to the parser and a [Set](#) gets created.

##### Parameters

<i>string</i>	set string that is to be parsed
---------------	---------------------------------

##### Returns

Null pointer if error occurs or a relation is returned

[Set](#) pointer when parsing is successful

#### 8.2.1.10 Set\* iegenlib::parser::parse\_set ( string set\_string )

parse\_set passes a string representation of a set to the parser and a ParseSet gets created.

##### Parameters

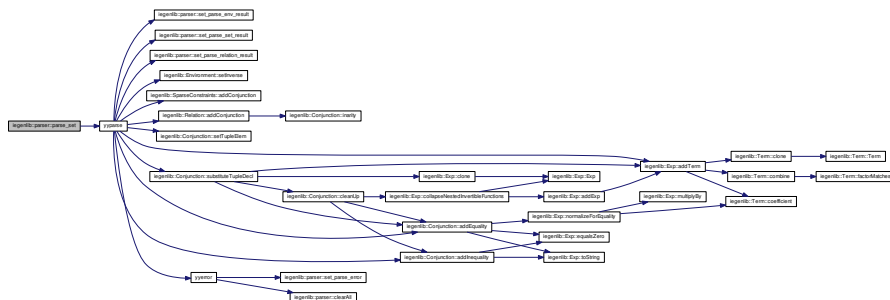
<i>string</i>	set string that is to be parsed
---------------	---------------------------------



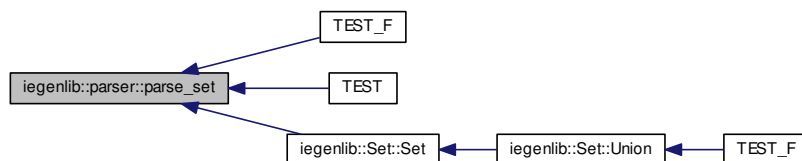
## Returns

Null pointer if error occurs or a relation is returned  
[Set](#) pointer when parsing is successful

Here is the call graph for this function:



Here is the caller graph for this function:



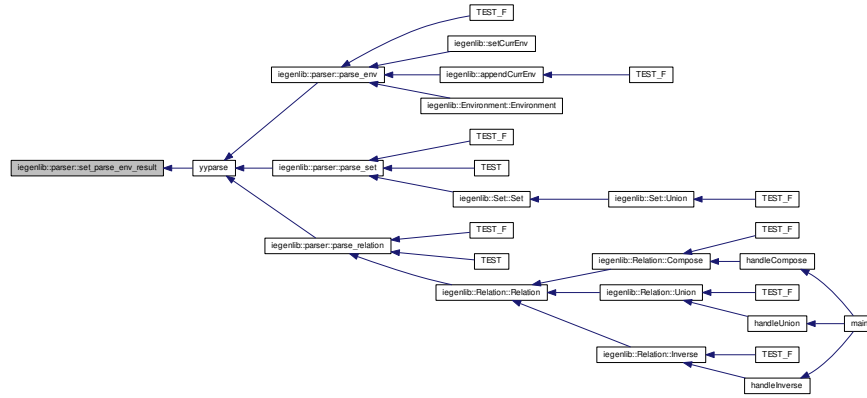
## 8.2.1.11 void iegenlib::parser::set\_parse\_env\_result ( Environment \* e )

setter for the `parse_env_result`

## Parameters

<a href="#">Environment</a>	<code>parse_env_result</code>
-----------------------------	-------------------------------

Here is the caller graph for this function:



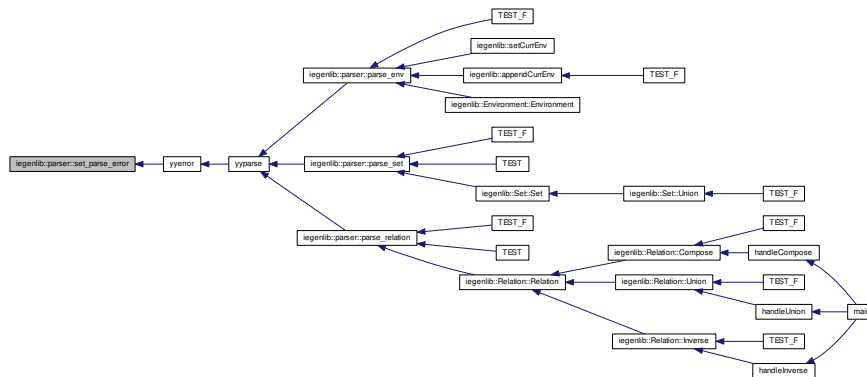
#### 8.2.1.12 void iegenlib::parser::set\_parse\_error ( std::string err )

sets the parse\_error flag to true

#### 8.2.1.13 void iegenlib::parser::set\_parse\_error ( string error )

sets the parse\_error flag to true

Here is the caller graph for this function:



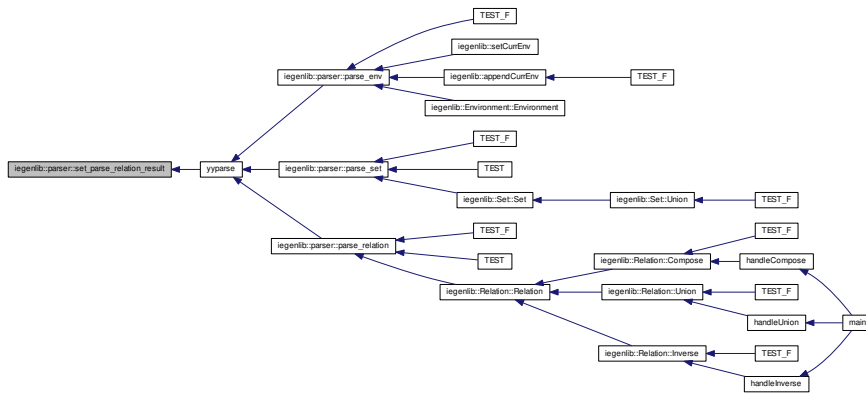
8.2.1.14 void iegenlib::parser::set\_parse\_relation\_result ( Relation \* s )

setter for the parse\_relation\_result

Parameters

<a href="#">Relation</a>	parse_relation_result
--------------------------	-----------------------

Here is the caller graph for this function:



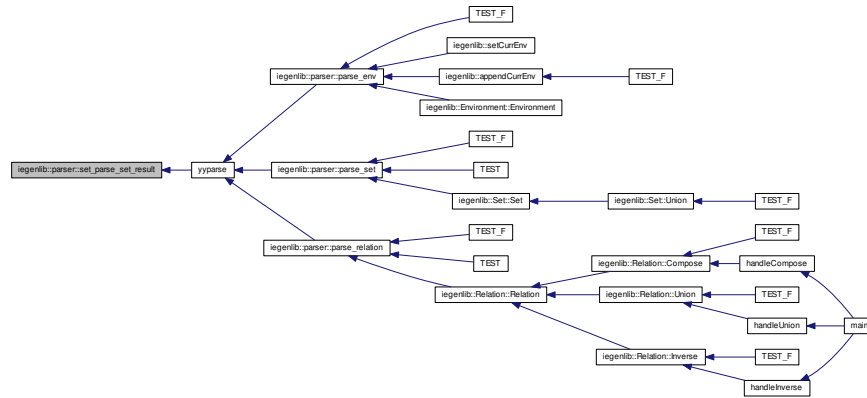
8.2.1.15 void iegenlib::parser::set\_parse\_set\_result ( Set \* s )

setter for the parse\_set\_result

Parameters

<a href="#">Set</a>	parse_set_result
---------------------	------------------

Here is the caller graph for this function:



### 8.2.1.16 int iegenlib::parser::string\_get\_next\_char ( )

Used by lexer to obtain each character of input string being parsed.

This is used for the actual parsing of the string we pass in we return each char of the string

#### Returns

int the value of each char of our string

## 8.2.2 Variable Documentation

### 8.2.2.1 string iegenlib::parser::error\_message

error message for when an error occurs in the parser

### 8.2.2.2 string iegenlib::parser::input\_buffer

a string of input buffer

### 8.2.2.3 unsigned int iegenlib::parser::input\_pos

the integer of our input position

**8.2.2.4 Environment\* iegenlib::parser::parse\_env\_result**

[Environment](#) pointer parse\_env\_result.

**8.2.2.5 bool iegenlib::parser::parse\_error**

boolean flag set when an error occurs in the parser

**8.2.2.6 Relation\* iegenlib::parser::parse\_relation\_result**

[Relation](#) pointer parse\_relation\_result.

**8.2.2.7 Set\* iegenlib::parser::parse\_set\_result**

[Set](#) pointer parse\_set\_result.



## Chapter 9

# Class Documentation

### 9.1 iegenlib::Conjunction Class Reference

Class containing sets of all the equalities and inequalities.

```
#include <set_relation.h>
```

#### Public Member Functions

- [Conjunction](#) (int arity)
- [Conjunction](#) (int arity, int inarity)
- [Conjunction](#) (const [Conjunction](#) &other)
- [Conjunction](#) & [operator=](#) (const [Conjunction](#) &other)
- virtual [~Conjunction](#) ()
- [Conjunction](#) \* [clone](#) () const  
*Create a copy of this Conjunction (of the same subclass)*
- void [setTupleElem](#) (int location, int constVal)
- void [setTupleElem](#) (int location, std::string varString)
- void [addEquality](#) ([Exp](#) \*equality)
- void [addInequality](#) ([Exp](#) \*inequality)
- const std::list< [Exp](#) \* > & [equalities](#) () const
- const std::list< [Exp](#) \* > & [inequalities](#) () const
- void [substituteTupleDecl](#) ()
- void [copyConstraintsFrom](#) (const [Conjunction](#) \*source)
- bool [substituteInConstraints](#) (const [Exp](#) \*sub, const [Term](#) &search)
- virtual std::string [toString](#) () const  
*Convert to a human-readable string.*
- virtual std::string [prettyPrintString](#) () const

*Convert to a human-readable string, pretty printed.*

- virtual std::string [toDotString](#) (int &next\_id) const
- virtual std::string [toDotString](#) (int parent\_id, int &next\_id) const
- int [arity](#) () const

*Get our arity.*

- int [inarity](#) () const

*Get inarity, for use with relations.*

- [Exp](#) \* [findFunction](#) (int tupleLocToFind, int startTupleRange, int endTupleRange) const
- [Exp](#) \* [findAndRemoveFunction](#) (int tupleLocToFind, int startTupleRange, int endTupleRange)
- [Conjunction](#) \* [Compose](#) (const [Conjunction](#) \*rhs, int innerArity) const
- [Conjunction](#) \* [Apply](#) (const [Conjunction](#) \*rhs) const
- [Conjunction](#) \* [Inverse](#) () const

*Compute Inverse of this conjunction.*

- bool [satisfiable](#) () const

*Return true if the constraints in the conjunction are satisfiable.*

- void [remapTupleVars](#) (const std::vector< int > &oldToNewLocs)
- void [cleanUp](#) ()

## Private Member Functions

- bool [isFunction](#) (int inArity) const
- bool [isFunctionInverse](#) (int inArity) const

## Private Attributes

- std::list< [Exp](#) \* > [mEqualities](#)  
*Set of equality constraints.*
- std::list< [Exp](#) \* > [mInequalities](#)  
*Set of inequality constraints.*
- std::vector< [TupleElemDecl](#) > [mTupleDecl](#)  
*List of the values in the tuple var associated with this conjunction.*
- int [mInArity](#)  
*To track how many tuple variables counted in the arity are input.*

### 9.1.1 Detailed Description

Class containing sets of all the equalities and inequalities.

This class holds two sets: one for the equalities and one for the inequalities.

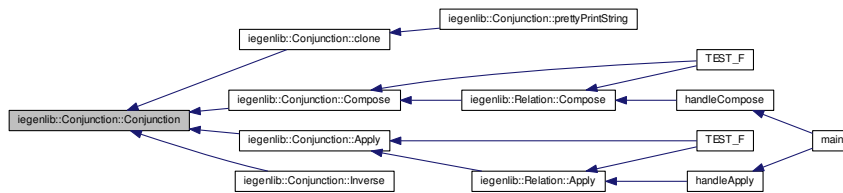


## 9.1.2 Constructor & Destructor Documentation

### 9.1.2.1 iegenlib::Conjunction::Conjunction ( int *arity* )

Constructor -- pass in the arity, which is immutable.

Here is the caller graph for this function:



### 9.1.2.2 iegenlib::Conjunction::Conjunction ( int *arity*, int *inarity* )

### 9.1.2.3 iegenlib::Conjunction::Conjunction ( const Conjunction & *other* )

### 9.1.2.4 iegenlib::Conjunction::~~Conjunction ( ) [virtual]

## 9.1.3 Member Function Documentation

### 9.1.3.1 void iegenlib::Conjunction::addEquality ( Exp \* *equality* )

addEquality -- add the given expression, interpreted as an equality ( $\text{Exp} = 0$ ), to our set of equalities.

#### Parameters

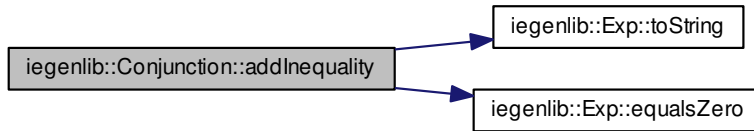
<i>equality</i>	(adopted)
-----------------	-----------

addEquality -- add the given expression, interpreted as an equality ( $\text{Exp} = 0$ ), to our list of equalities. Maintains a sorted order on the constraints.

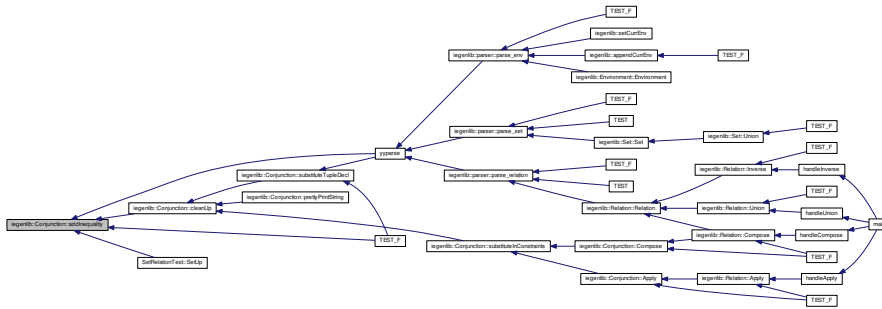
FIXME: the check for duplicate equalities is done in  $O(n)$  time here, but with a different strategy could be reduced to  $O(\log n)$  time.



Here is the call graph for this function:



Here is the caller graph for this function:



### 9.1.3.3 Conjunction \* iegenlib::Conjunction::Apply ( const Conjunction \* rhs ) const

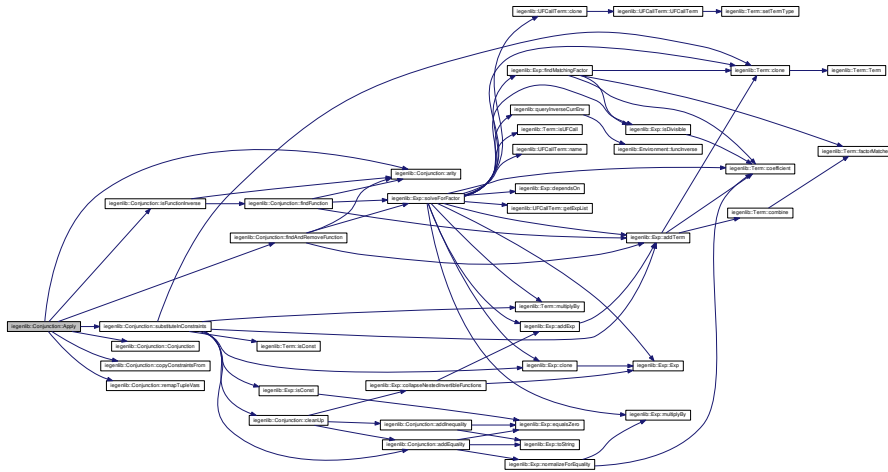
Apply this (interpreted as a [Relation](#)) to rhs, which is interpreted as a set.  $r = \{ x \rightarrow y : x = G(y) \ \&\& \ C \}$   $s = \{ z : D \}$   $r(s) = \{ y : D[G(y)] \ \&\& \ C[G(y)] \}$

Parameters

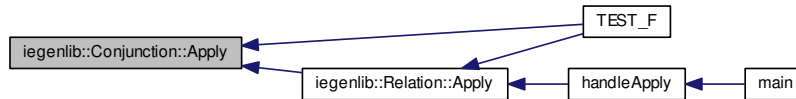
<i>rhs</i>	(not adopted)
------------	---------------

Apply this (interpreted as a [Relation](#)) to rhs, which is interpreted as a set.

Here is the call graph for this function:



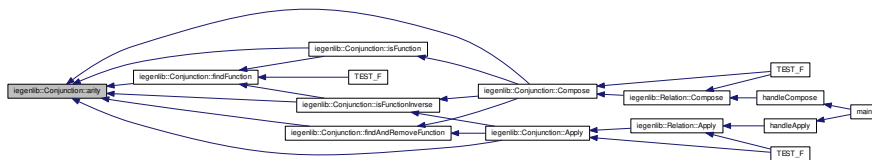
Here is the caller graph for this function:



### 9.1.3.4 int ieigenlib::Conjunction::arity ( ) const [inline]

Get our arity.

Here is the caller graph for this function:

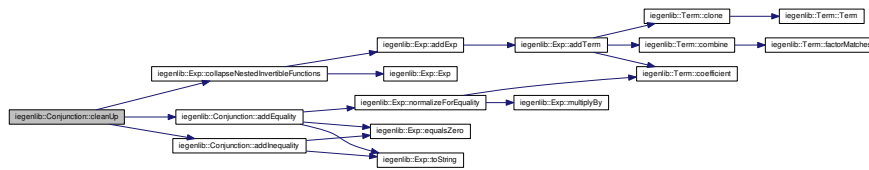


9.1.3.5 void igenlib::Conjunction::cleanUp ( )

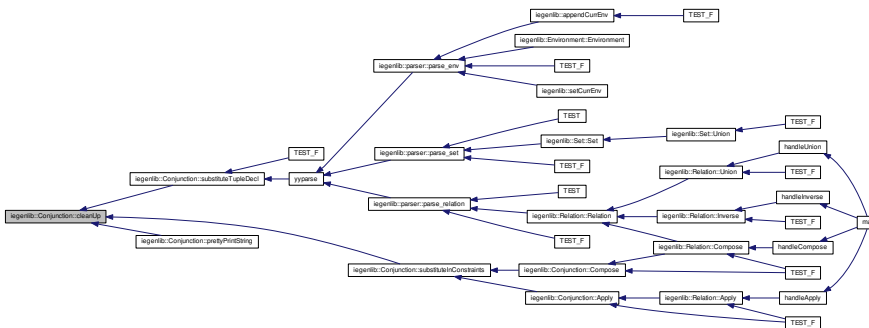
Cleans up constraints in the conjunction.

- Removes equality and inequality constraints that are equal to zero from list of Expressions.
- Also removes duplicate constraints.

Removes equality and inequality Exps that are equal to zero from list of Expressions. Here is the call graph for this function:



Here is the caller graph for this function:



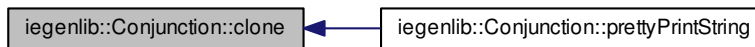
9.1.3.6 Conjunction \* igenlib::Conjunction::clone ( ) const

Create a copy of this Conjunction (of the same subclass)

Here is the call graph for this function:



Here is the caller graph for this function:



#### 9.1.3.7 `Conjunction * iegenlib::Conjunction::Compose ( const Conjunction * rhs, int innerArity ) const`

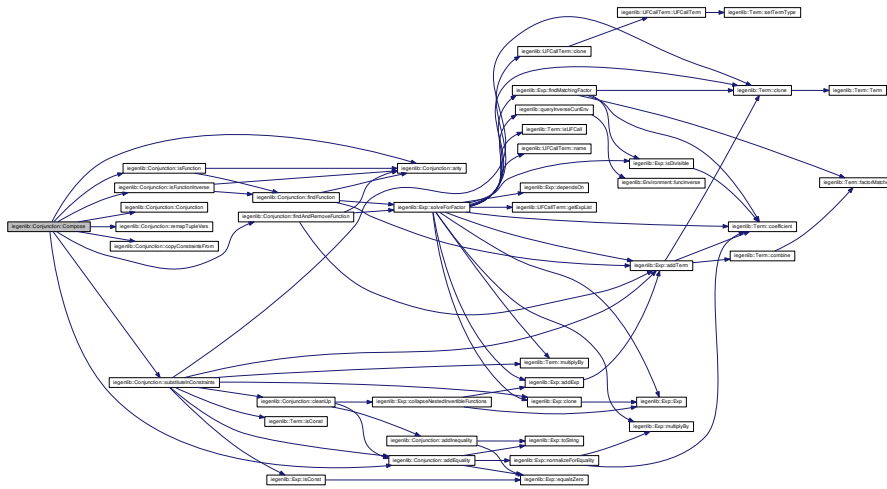
Compose with another conjunction, given innerArity (which is this's in arity and rhs's out arity).

##### Parameters

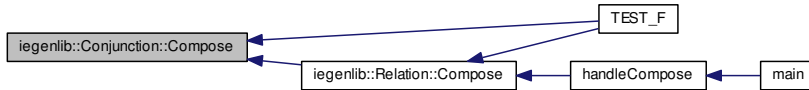
<i>rhs</i>	(not adopted)
<i>innerArity</i>	

Compose with another conjunction, given innerArity (which is this's in arity and rhs's out arity).

Here is the call graph for this function:



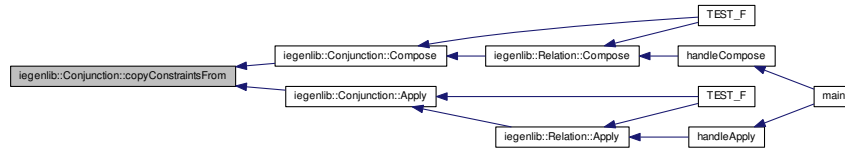
Here is the caller graph for this function:



9.1.3.8 void iegenlib::Conjunction::copyConstraintsFrom ( const Conjunction \* source )

copyConstraintsFrom -- copy all the equalities and inequalities from source, and add them to our own constraints.

Here is the caller graph for this function:



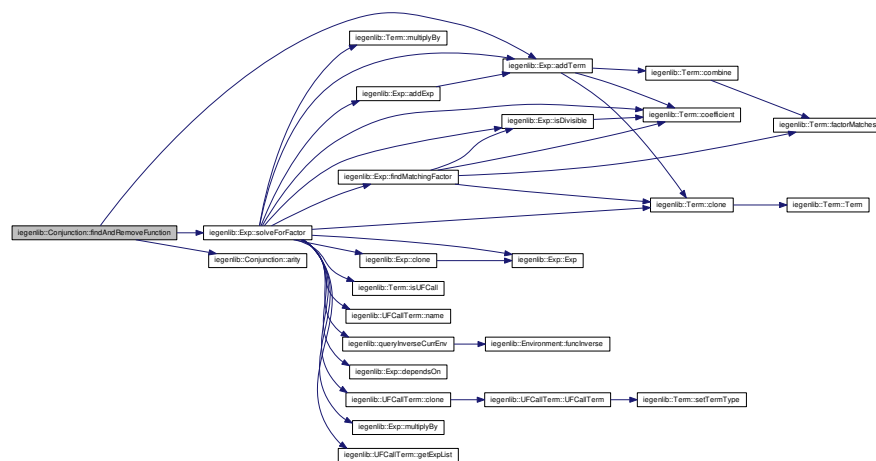
9.1.3.9 `const std::list<Exp*> & iegenlib::Conjunction::equalities ( ) const` [inline]

9.1.3.10 `Exp * iegenlib::Conjunction::findAndRemoveFunction ( int tupleLocToFind, int startTupleRange, int endTupleRange )`

Same as findFunction except that the equality the function is derived from is removed.

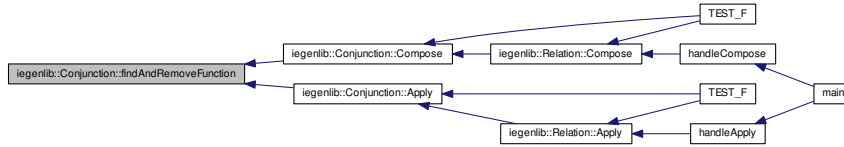
Search among our equality constraints for one that defines tupleLocToFind as a function of only tuple variables in the location range [startTupleRange, endTupleRange]. Return a new copy of that expression. Additionally remove the equality constraint that the function was derived from.

Here is the call graph for this function:





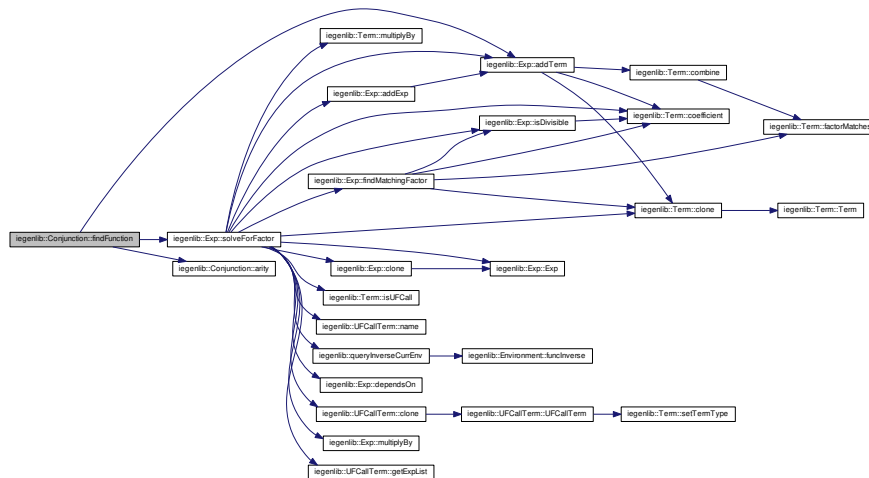
Here is the caller graph for this function:



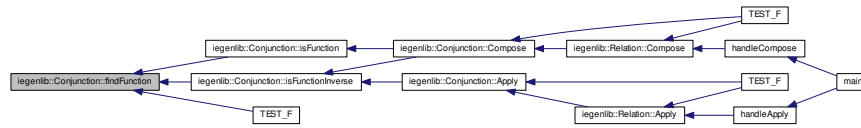
9.1.3.11 `Exp * iegenlib::Conjunction::findFunction ( int tupleLocToFind, int startTupleRange, int endTupleRange ) const`

Search among our equality constraints for one that defines `tupleLocToFind` as a function of only tuple variables in the location range `[startTupleRange, endTupleRange]`. Return a new copy of that expression.

Here is the call graph for this function:



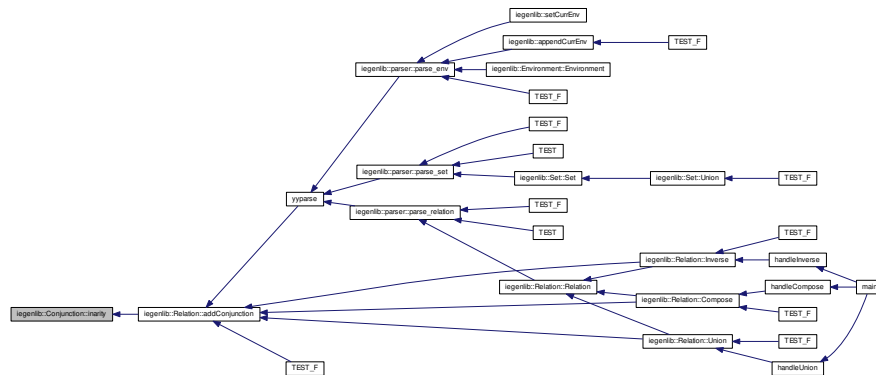
Here is the caller graph for this function:



### 9.1.3.12 `int iegenlib::Conjunction::inarity ( ) const [inline]`

Get inarity, for use with relations.

Here is the caller graph for this function:



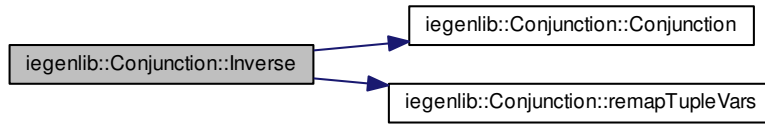
### 9.1.3.13 `const std::list<Exp*>& iegenlib::Conjunction::inequalities ( ) const [inline]`

### 9.1.3.14 `Conjunction * iegenlib::Conjunction::Inverse ( ) const`

Compute Inverse of this conjunction.

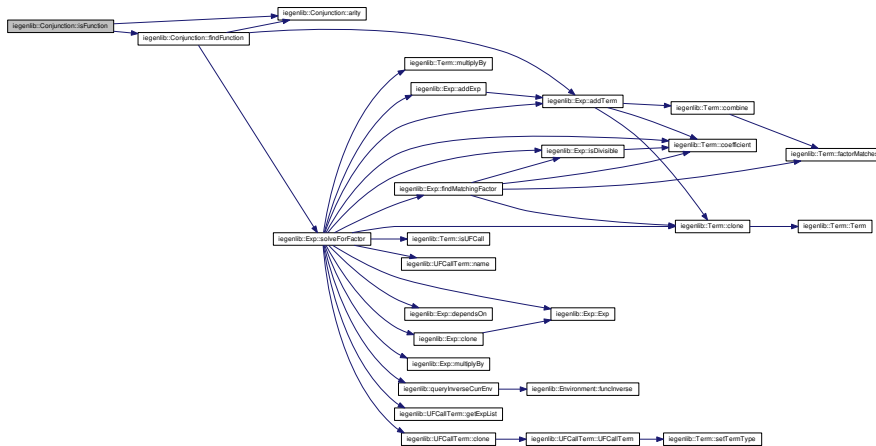
Create the inverse of this conjunction. Returns a new [Conjunction](#), which the caller is responsible for deallocating.

Here is the call graph for this function:

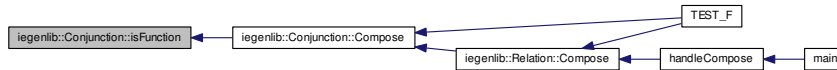


9.1.3.15 bool iegenlib::Conjunction::isFunction ( int inAriety ) const [private]

Here is the call graph for this function:

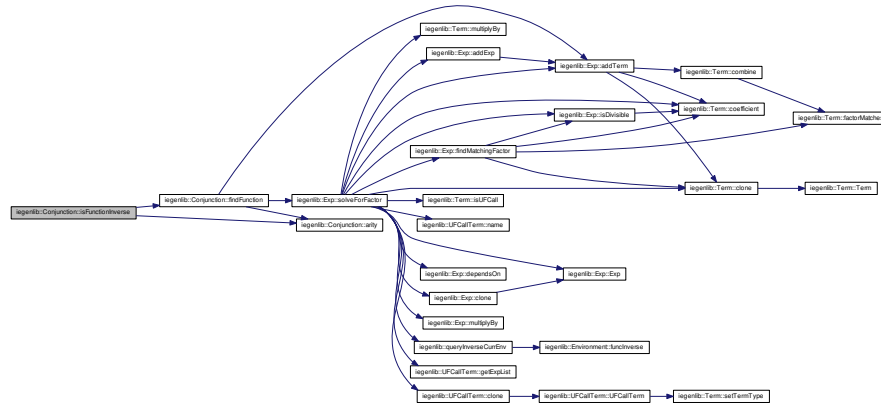


Here is the caller graph for this function:

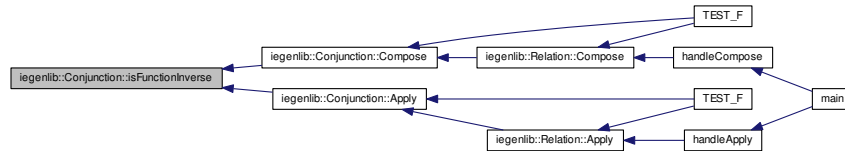


### 9.1.3.16 `bool iegenlib::Conjunction::isFunctionInverse ( int inArity ) const` [private]

Here is the call graph for this function:



Here is the caller graph for this function:



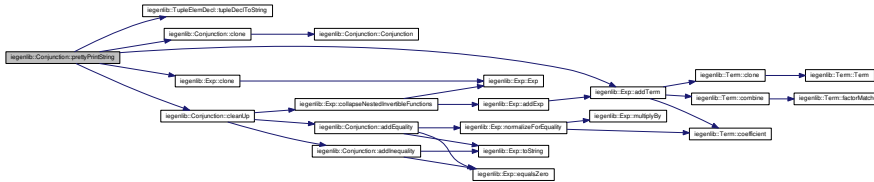
### 9.1.3.17 `Conjunction & iegenlib::Conjunction::operator= ( const Conjunction & other )`

### 9.1.3.18 `std::string iegenlib::Conjunction::prettyPrintString ( ) const` [virtual]

Convert to a human-readable string, pretty printed.

Convert to a human-readable string (sub in for tuple vars).

Here is the call graph for this function:

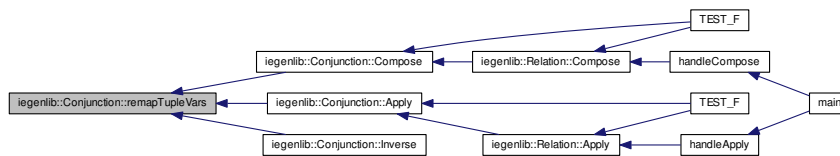


### 9.1.3.19 void iegenlib::Conjunction::remapTupleVars ( const std::vector< int > & oldToNewLocs )

Find any TupleVarTerms in this expression (and subexpressions) and remap the locations according to the oldToNewLocs vector, where oldToNewLocs[i] = j means that old location i becomes new location j (i.e. `__tvi -> __tvj`). Throws an exception if an old location is out of range for the given oldToNewLocs.

Find any TupleVarTerms in this expression (and subexpressions) and remap the locations according to the oldToNewLocs vector, where oldToNewLocs[i] = j means that old location i becomes new location j (i.e. `__tvi -> __tvj`). As a special case, a value of -1 means that old location goes away entirely.

Here is the caller graph for this function:



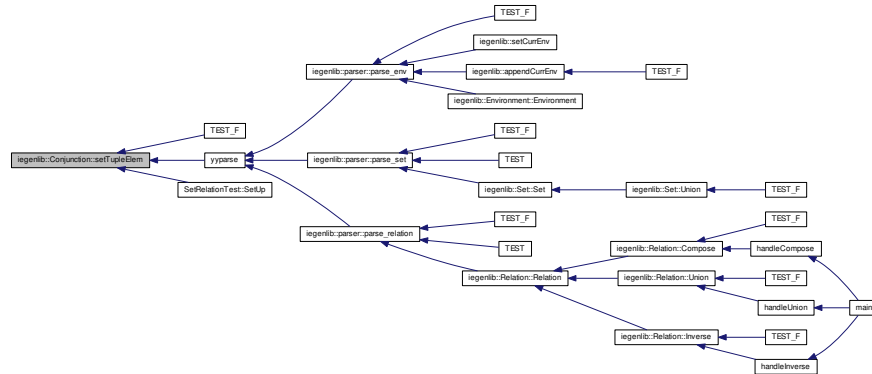
### 9.1.3.20 bool iegenlib::Conjunction::satisfiable ( ) const

Return true if the constraints in the conjunction are satisfiable.

### 9.1.3.21 void iegenlib::Conjunction::setTupleElem ( int *location*, int *constVal* )

setTupleElem -- defines a tuple element as either a constant value, or a variable with the given name.

Here is the caller graph for this function:



### 9.1.3.22 void iegenlib::Conjunction::setTupleElem ( int *location*, std::string *varString* )

### 9.1.3.23 bool iegenlib::Conjunction::substituteInConstraints ( const Exp \* *sub*, const Term & *search* )

Substitute an expression for a factor (i.e. the non-coefficient part of a term) in all our equalities and inequalities. Handles cases where one or both of the parameters are constants.

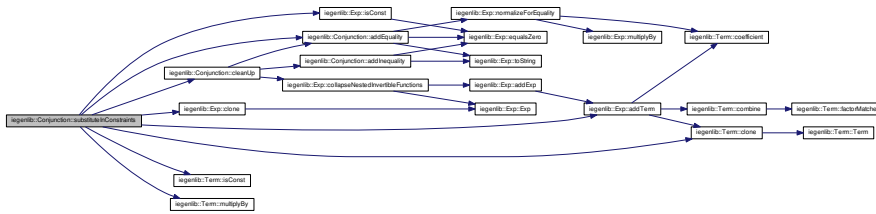
#### Parameters

<i>sub</i>	(not adopted)
<i>search</i>	(not adopted)

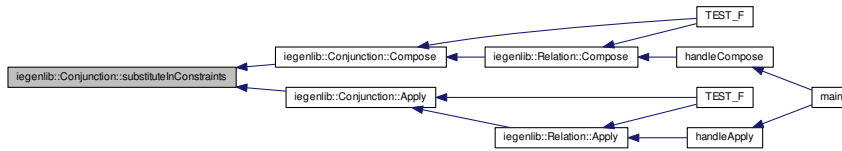
Returns

true if substitution was successful, false if substituting a constant for a different constant

Here is the call graph for this function:



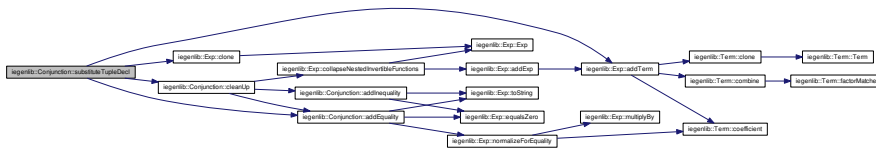
Here is the caller graph for this function:



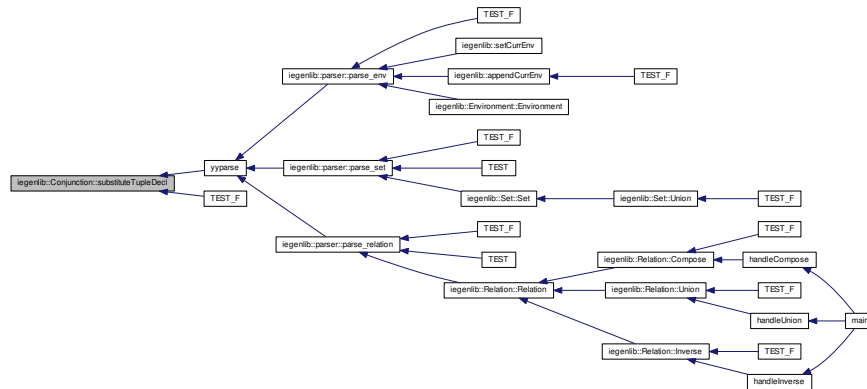
9.1.3.24 void iegenlib::Conjunction::substituteTupleDecl ( )

substituteTupleDecl -- substitute TupleVarTerms in for any VarTerms in the expressions whose names match the corresponding tuple element declaration.

Here is the call graph for this function:



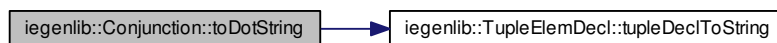
Here is the caller graph for this function:



#### 9.1.3.25 `std::string iegenlib::Conjunction::toDotString ( int & next_id ) const [virtual]`

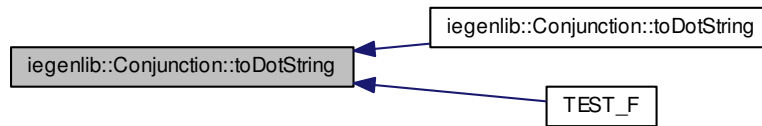
Convert to a DOT string. Pass in the parent node id and the next node id. The next node id will be set upon exit from this routine. If no parent id is given then will not draw edge from parent to self.

Here is the call graph for this function:



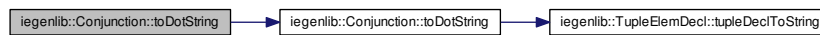


Here is the caller graph for this function:



9.1.3.26 `std::string iegenlib::Conjunction::toDotString ( int parent_id, int & next_id ) const` [virtual]

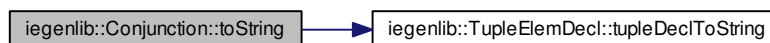
Here is the call graph for this function:



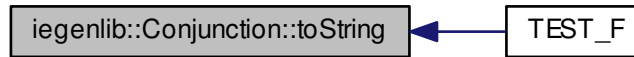
9.1.3.27 `std::string iegenlib::Conjunction::toString ( ) const` [virtual]

Convert to a human-readable string.

Here is the call graph for this function:



Here is the caller graph for this function:



## 9.1.4 Member Data Documentation

9.1.4.1 `std::list<Exp*> iegenlib::Conjunction::mEqualities` [private]

Set of equality constraints.

9.1.4.2 `int iegenlib::Conjunction::mInArity` [private]

To track how many tuple variables counted in the arity are input.

9.1.4.3 `std::list<Exp*> iegenlib::Conjunction::mInequalities` [private]

Set of inequality constraints.

9.1.4.4 `std::vector<TupleElemDecl> iegenlib::Conjunction::mTupleDecl`  
[private]

List of the values in the tuple var associated with this conjunction.

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

- [src/iegenlib/set\\_relation/set\\_relation.h](#)
- [src/iegenlib/set\\_relation/set\\_relation.cc](#)

## 9.2 iegenlib::Environment Class Reference

```
#include <environment.h>
```

## Public Member Functions

- [Environment](#) ()  
*Environment constructor.*
- [Environment](#) (std::string str)  
*Construct an environment and parse the string as the first members.*
- [Environment](#) (const [Environment](#) &other)  
*Copy constructor for Environment.*
- [Environment](#) & operator= (const [Environment](#) &other)  
*Assignment operator for Environment.*
- void [append](#) ([Environment](#) \*other)  
*append an environment to this one (adopts)*
- void [reset](#) ()  
*Reset the Environment to empty.*
- void [setInverse](#) (std::string funcName, std::string inverseName)  
*Define the inverse for the given function.*
- std::string [funcInverse](#) (std::string funcName) const  
*Get the name of the inverse of the given function (or "" if none).*
- bool [hasInverse](#) (const std::string funcName) const  
*Check whether the given function has a known inverse.*
- std::string [toString](#) () const

## Private Attributes

- std::map< std::string, std::string > [mInverseMap](#)

### 9.2.1 Constructor & Destructor Documentation

#### 9.2.1.1 iegenlib::Environment::Environment ( ) [inline]

[Environment](#) constructor.

#### 9.2.1.2 iegenlib::Environment::Environment ( std::string str )

Construct an environment and parse the string as the first members.

Construct an [Environment](#).



Here is the caller graph for this function:



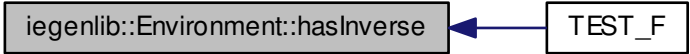
9.2.2.3 bool iegenlib::Environment::hasInverse ( const std::string *funcName* ) const  
[inline]

Check whether the given function has a known inverse.

Here is the call graph for this function:



Here is the caller graph for this function:



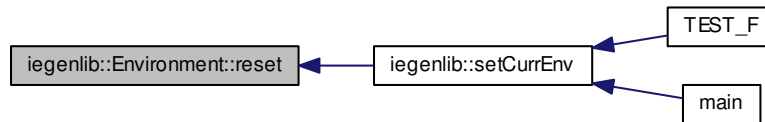
9.2.2.4 Environment & iegenlib::Environment::operator= ( const Environment & *other* )

Assignment operator for [Environment](#).

## 9.2.2.5 void iegenlib::Environment::reset ( )

Reset the [Environment](#) to empty.

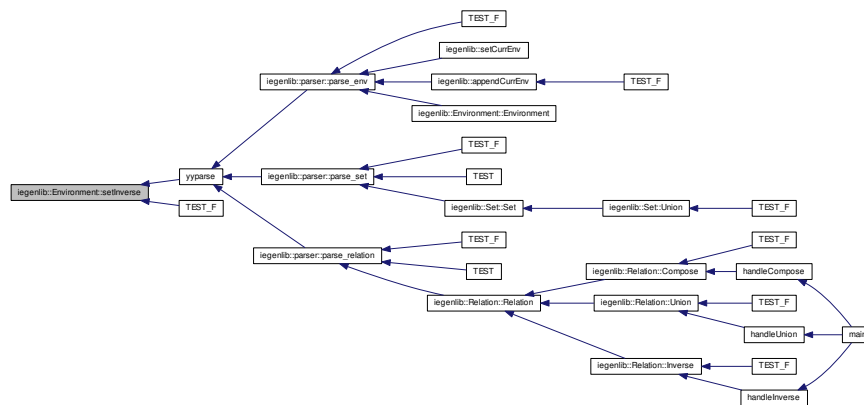
Here is the caller graph for this function:



## 9.2.2.6 void iegenlib::Environment::setInverse ( std::string funcName, std::string inverseName )

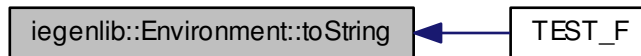
Define the inverse for the given function.

Here is the caller graph for this function:



### 9.2.2.7 `std::string iegenlib::Environment::toString ( ) const`

Here is the caller graph for this function:



## 9.2.3 Member Data Documentation

### 9.2.3.1 `std::map<std::string, std::string> iegenlib::Environment::mInverseMap` [private]

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

- [src/iegenlib/set\\_relation/environment.h](#)
- [src/iegenlib/set\\_relation/environment.cc](#)

## 9.3 EnvironmentTest Class Reference

### Protected Member Functions

- void [SetUp](#) ( )
- void [TearDown](#) ( )

### 9.3.1 Member Function Documentation

#### 9.3.1.1 `void EnvironmentTest::SetUp ( )` [inline, protected]

#### 9.3.1.2 `void EnvironmentTest::TearDown ( )` [inline, protected]

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

- [src/iegenlib/set\\_relation/environment\\_test.cc](#)

## 9.4 ExceptionTestExp Class Reference

Class to test exception handling in expression/ methods.

### Protected Member Functions

- virtual void [SetUp](#) ()
- virtual void [TearDown](#) ()

#### 9.4.1 Detailed Description

Class to test exception handling in expression/ methods.

This class holds gtest test cases that test the functionality of the methods inside [expression.cc](#) in regards to exception handling.

#### 9.4.2 Member Function Documentation

9.4.2.1 void `ExceptionTestExp::SetUp` ( ) [protected, virtual]

9.4.2.2 void `ExceptionTestExp::TearDown` ( ) [protected, virtual]

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

- [src/iegenlib/exceptions\\_test.cc](#)

## 9.5 ExceptionTestParser Class Reference

Class to test exception handling in parser/ methods.

### Protected Member Functions

- virtual void [SetUp](#) ()
- virtual void [TearDown](#) ()

#### 9.5.1 Detailed Description

Class to test exception handling in parser/ methods.

This class holds gtest test cases that test the functionality of the methods inside [parser.-cc](#) in regards to exception handling.



## 9.5.2 Member Function Documentation

9.5.2.1 void ExceptionTestParser::SetUp ( ) [protected, virtual]

9.5.2.2 void ExceptionTestParser::TearDown ( ) [protected, virtual]

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

- [src/iegenlib/exceptions\\_test.cc](#)

## 9.6 ExceptionTestSetRelation Class Reference

Class to test exception handling in set\_relation/ methods.

### Protected Member Functions

- virtual void [SetUp](#) ( )
- virtual void [TearDown](#) ( )

### 9.6.1 Detailed Description

Class to test exception handling in set\_relation/ methods.

This class holds gtest test cases that test the functionality of the methods inside [set\\_relation.cc](#) in regards to exception handling.

### 9.6.2 Member Function Documentation

9.6.2.1 void ExceptionTestSetRelation::SetUp ( ) [protected, virtual]

9.6.2.2 void ExceptionTestSetRelation::TearDown ( ) [protected, virtual]

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

- [src/iegenlib/exceptions\\_test.cc](#)

## 9.7 iegenlib::Exp Class Reference

An affine expression that allows uninterpreted function call terms.

```
#include <expression.h>
```

## Public Member Functions

- [Exp](#) ()  
*Default constructor.*
- [Exp](#) (const [Exp](#) &other)  
*Copy constructor. Performs a deep copy.*
- [Exp](#) & [operator=](#) (const [Exp](#) &other)  
*Copy assignment.*
- virtual [~Exp](#) ()  
*Destructor.*
- virtual [Exp](#) \* [clone](#) () const  
*Create a copy of this [Exp](#) (and of the same subclass)*
- virtual std::string [toString](#) () const  
*Creates a compact string to help with debugging.*
- virtual std::string [prettyPrintString](#) (const std::vector< [TupleElemDecl](#) > &m-TupleDecl) const  
*Convert to a human-readable string (substitute in tuple vars).*
- void [addTerm](#) ([Term](#) \*term)  
*Add a term to this expression.*
- void [addExp](#) ([Exp](#) \*exp)  
*Add another expression to this one.*
- void [multiplyBy](#) (int constant)  
*Multiply all terms in this expression by a constant.*
- bool [isDivisible](#) (int divisor) const
- void [divideBy](#) (int divisor)  
*Divide all coefficients and the constant term by the given divisor.*
- [Exp](#) \* [solveForFactor](#) ([Term](#) \*factor) const
- void [substitute](#) ([Exp](#) \*sub, const [Term](#) &search)
- void [normalizeForEquality](#) ()
- [Exp](#) \* [collapseNestedInvertibleFunctions](#) () const
- bool [dependsOn](#) (const [Term](#) &factor) const
- bool [equalsZero](#) () const
- bool [operator==](#) (const [Term](#) &other) const  
*Returns true if this expression equals the given term.*
- void [remapTupleVars](#) (const std::vector< int > &oldToNewLocs)
- void [setExpression](#) ()  
*Calls the [ExpCase](#) for the visitor design pattern.*
- void [setInequality](#) ()  
*Sets [mExpType](#) to [Inequality](#), to indicate [Exp](#) >= 0.*
- void [setEquality](#) ()  
*Sets [mExpType](#) to [Equality](#), to indicate [Exp](#) == 0.*

- bool `isExpression ()`  
*Returns true if the `Exp` is a simple expression.*
- bool `isInequality ()`  
*Returns true if the `Exp` is an inequality, ie expression  $\geq 0$ .*
- bool `isEquality ()`  
*Returns true if the `Exp` is an equality, ie expression  $= 0$ .*
- bool `isConst () const`  
*Returns true if only have a constant term.*
- `Term * getTerm () const`
- `std::string toDotString (int &next_id) const`
- `std::string toDotString (int parent_id, int &next_id) const`

### Protected Types

- enum `exptype { Expression, Inequality, Equality }`

### Protected Member Functions

- `exptype getExpType () const`

### Private Member Functions

- `Term * findMatchingFactor (const Term &factor) const`

### Private Attributes

- `std::list< Term * > mTerms`
- `exptype mExpType`

#### 9.7.1 Detailed Description

An affine expression that allows uninterpreted function call terms.

Memory management: the `Exp` manages its own copies of any terms it contains, copies those when the `Exp` itself is copied, and deletes them when the `Exp` is destroyed.

## 9.7.2 Member Enumeration Documentation

### 9.7.2.1 enum `iegenlib::Exp::exptype` [protected]

Enumerator:

***Expression***

***Inequality***

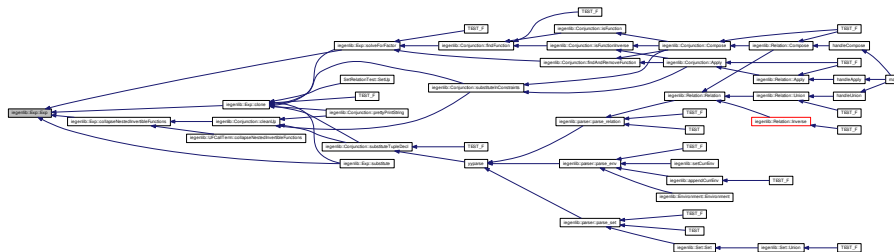
***Equality***

## 9.7.3 Constructor & Destructor Documentation

### 9.7.3.1 `iegenlib::Exp::Exp ( )` [inline]

Default constructor.

Here is the caller graph for this function:



### 9.7.3.2 `iegenlib::Exp::Exp ( const Exp & other )`

Copy constructor. Performs a deep copy.

Copy constructor.

### 9.7.3.3 `iegenlib::Exp::~Exp ( )` [virtual]

Destructor.

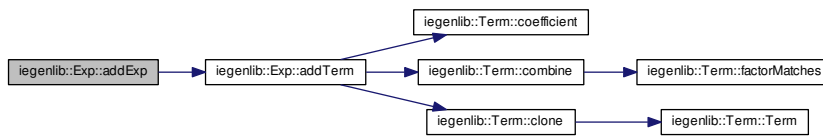
## 9.7.4 Member Function Documentation

9.7.4.1 void iegenlib::Exp::addExp ( Exp \* exp )

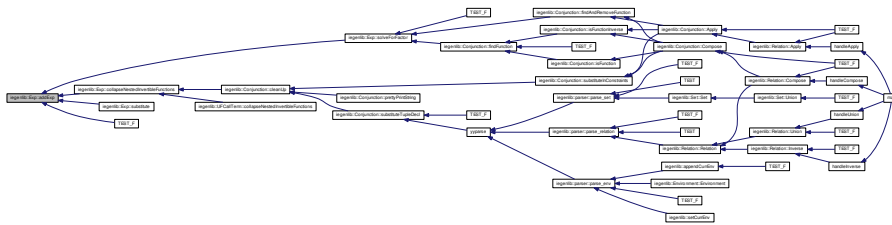
Add another expression to this one.

Add another expression to this one. /param term (adopted)

Here is the call graph for this function:



Here is the caller graph for this function:

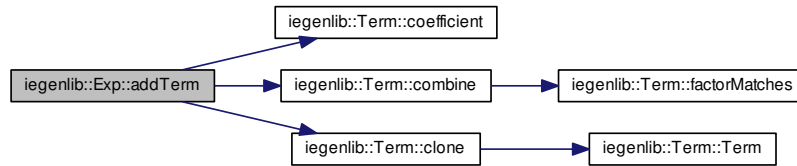


9.7.4.2 void iegenlib::Exp::addTerm ( Term \* term )

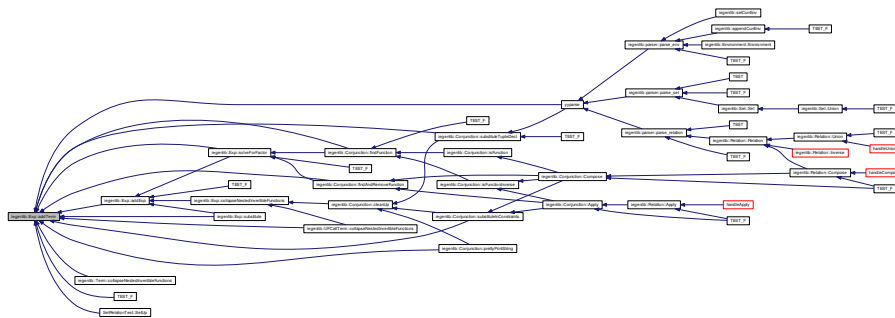
Add a term to this expression.

Add a term to this expression. /param term (adopted)

Here is the call graph for this function:



Here is the caller graph for this function:



### 9.7.4.3 `Exp * iegenlib::Exp::clone ( ) const [virtual]`

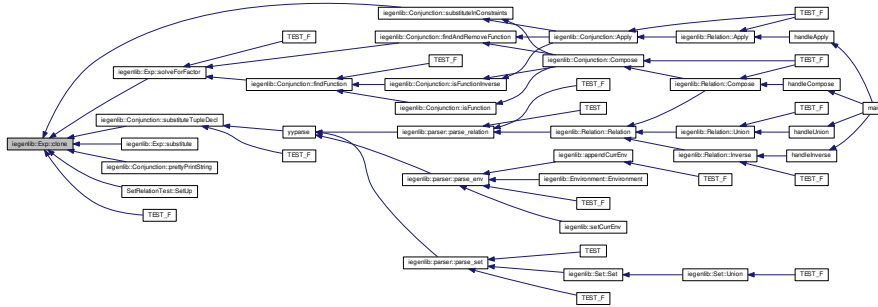
Create a copy of this [Exp](#) (and of the same subclass)

Create a copy of this [Exp](#) (of the same subclass)

Here is the call graph for this function:



Here is the caller graph for this function:



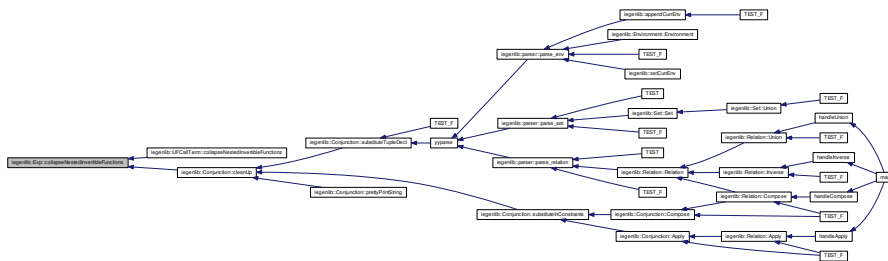
#### 9.7.4.4 Exp \* igenlib::Exp::collapseNestedInvertibleFunctions ( ) const

Return a new `Exp` with all nested functions such as  $f(f_{\text{inv}}(i))$  changed to  $i$ .

Here is the call graph for this function:



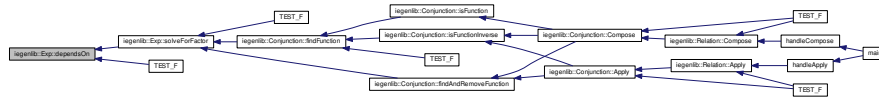
Here is the caller graph for this function:



#### 9.7.4.5 bool `iegenlib::Exp::dependsOn ( const Term & factor ) const`

Search for the given factor anywhere in this expression (including within [UFCallTerm](#) arguments, recursively).

Here is the caller graph for this function:



#### 9.7.4.6 void `iegenlib::Exp::divideBy ( int divisor )`

Divide all coefficients and the constant term by the given divisor.

Here is the call graph for this function:

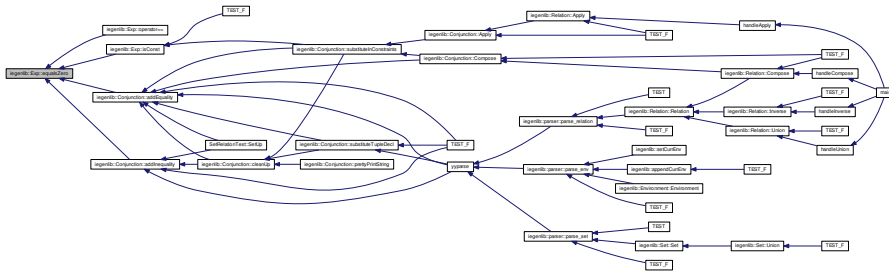


#### 9.7.4.7 bool `iegenlib::Exp::equalsZero ( ) const`

Return true iff this expression has no terms, or has only a constant term equal to 0.



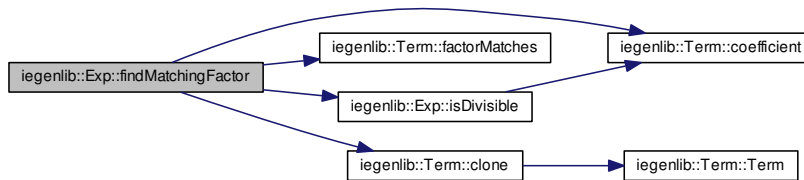
Here is the caller graph for this function:



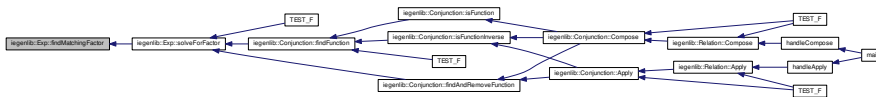
9.7.4.8 Term \* iegenlib::Exp::findMatchingFactor ( const Term & factor ) const  
[private]

Search this [Exp](#) for the given factor. The cloned [Term](#) that is returned can have a coefficient other than 1. The factor param should have a coefficient of 1. Returns NULL if a matching [Term](#) is not found.

Here is the call graph for this function:

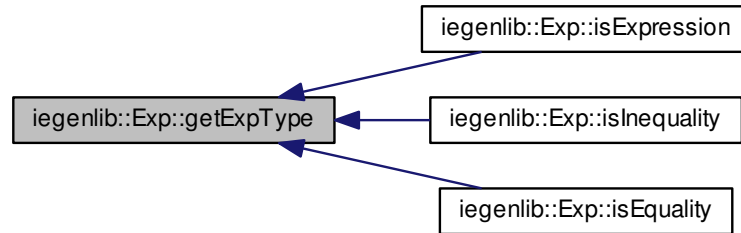


Here is the caller graph for this function:



#### 9.7.4.9 `exptype iegenlib::Exp::getExpType ( ) const` [inline, protected]

Here is the caller graph for this function:

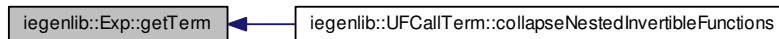


#### 9.7.4.10 `Term* iegenlib::Exp::getTerm ( ) const`

Return `Term*` if the expression has only one `Term`. Otherwise returns `NULL`. this still owns `Term`.

Return `Term*` if the expression has only one `Term`. Otherwise returns `NULL`. Still owns `Term`.

Here is the caller graph for this function:



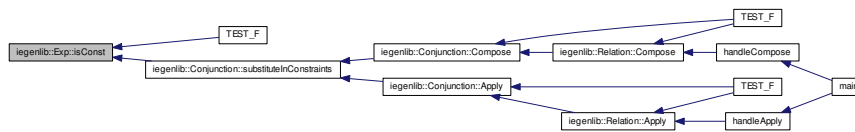
#### 9.7.4.11 `bool iegenlib::Exp::isConst ( ) const`

Returns true if only have a constant term.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 9.7.4.12 bool iegenlib::Exp::isDivisible ( int *divisor* ) const

Return whether all coefficients in this expression are evenly divisible by the given integer.

Here is the call graph for this function:





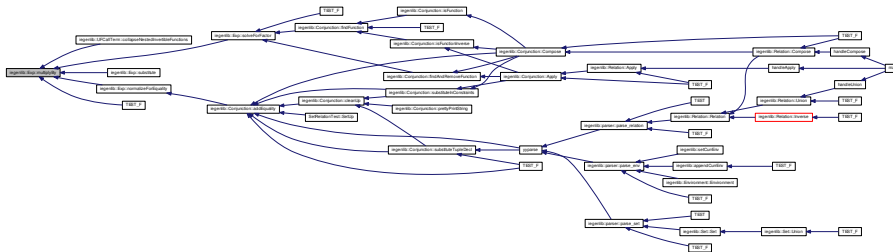
Here is the call graph for this function:



#### 9.7.4.16 void iegenlib::Exp::multiplyBy ( int *constant* )

Multiply all terms in this expression by a constant.

Here is the caller graph for this function:



#### 9.7.4.17 void iegenlib::Exp::normalizeForEquality ( )

Normalize this expression for use in an equality expression. This is called when we know this expression is equal to zero; in that case, it's valid to multiply the whole expression by -1. So we do so, in order to ensure the first term has a positive coefficient, so that equivalent expressions can be reliably compared.



Here is the call graph for this function:



9.7.4.20 `std::string iegenlib::Exp::prettyPrintString ( const std::vector< TupleElemDecl > & mTupleDecl ) const` [virtual]

Convert to a human-readable string (substitute in tuple vars).

Creates a compact string, pretty printed.

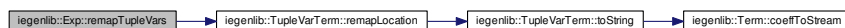
Here is the caller graph for this function:



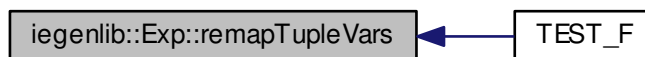
9.7.4.21 `void iegenlib::Exp::remapTupleVars ( const std::vector< int > & oldToNewLocs )`

Find any `TupleVarTerms` in this expression (and subexpressions) and remap the locations according to the `oldToNewLocs` vector, where `oldToNewLocs[i] = j` means that old location `i` becomes new location `j` (i.e. `__tvi -> __tvj`). Throws an exception if an old location is out of range for the given `oldToNewLocs`.

Here is the call graph for this function:



Here is the caller graph for this function:



9.7.4.22 void iegenlib::Exp::setEquality ( ) [inline]

Sets mExpType to Equality, to indicate `Exp == 0`.

9.7.4.23 void iegenlib::Exp::setExpression ( ) [inline]

Calls the ExpCase for the visitor design pattern.

Sets mExpType to Expression, for a simple expression

9.7.4.24 void iegenlib::Exp::setInequality ( ) [inline]

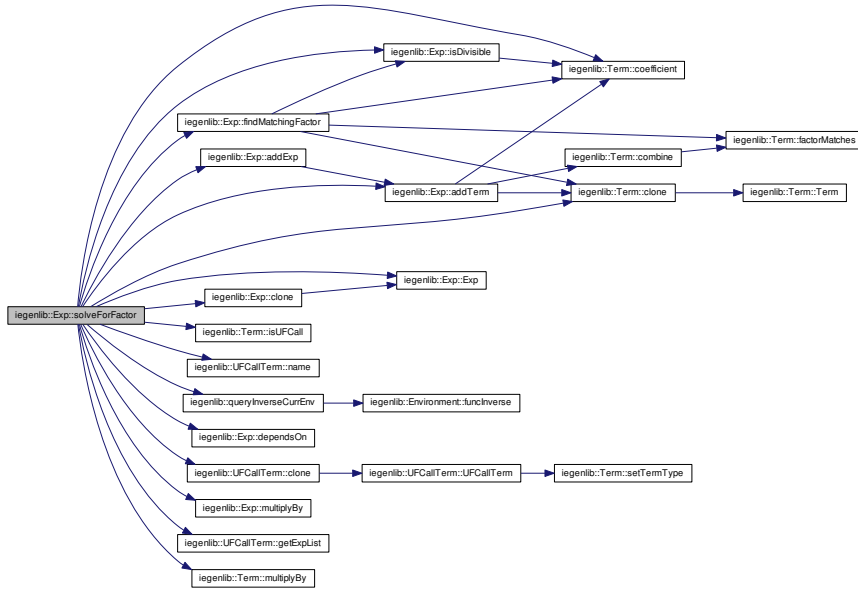
Sets mExpType to Inequality, to indicate `Exp >= 0`.

9.7.4.25 `Exp * iegenlib::Exp::solveForFactor ( Term * factor ) const`

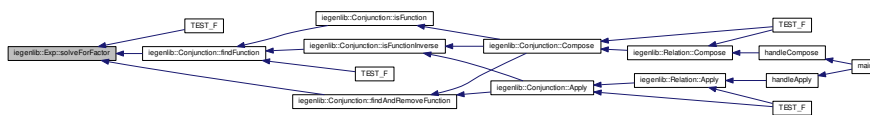
/param factor (adopted)



Here is the call graph for this function:



Here is the caller graph for this function:

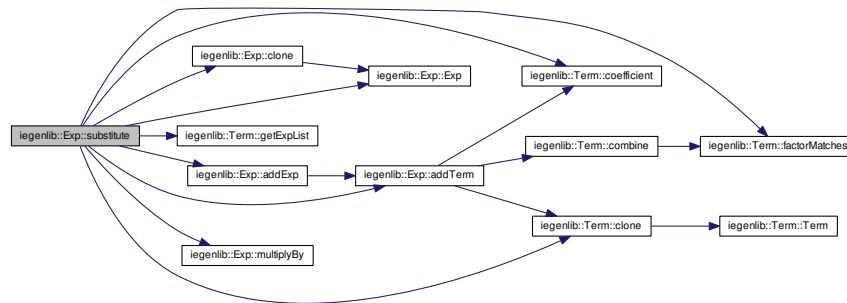


9.7.4.26 void iegenlib::Exp::substitute ( Exp \* sub, const Term & search )

Substitute an expression for a factor (i.e. the non-coefficient part of a term). /param sub (adopted) /param search (not adopted)

Substitute an expression for a factor (i.e. the non-coefficient part of a term).

Here is the call graph for this function:

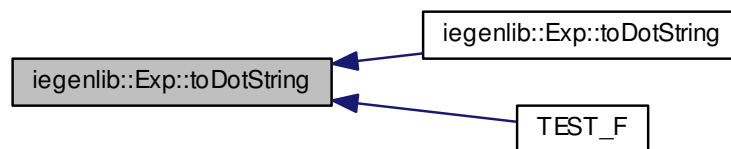


#### 9.7.4.27 `std::string iegenlib::Exp::toDotString ( int & next_id ) const`

Output the [Exp](#) in dot format. Note here, we still need to provide "digraph name {" and "}" Pass in the parent node id and the next node id. The next node id will be set upon exit from this routine. If no parent id is given then will not draw edge from parent to self.

Output the [Exp](#) in dot format. Pass in the next node id. The next node id will be set to next id upon exit from this routine.

Here is the caller graph for this function:



#### 9.7.4.28 `std::string iegenlib::Exp::toDotString ( int parent_id, int & next_id ) const`

Output the [Exp](#) in dot format. Pass in the parent node id and the next node id. The next node id will be set upon exit from this routine.

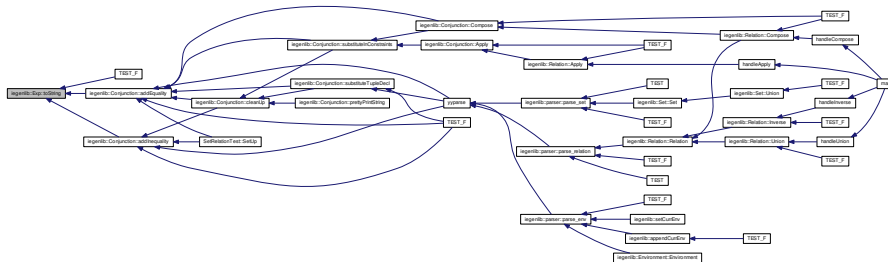
Here is the call graph for this function:



#### 9.7.4.29 `std::string iegenlib::Exp::toString ( ) const [virtual]`

Creates a compact string to help with debugging.

Here is the caller graph for this function:



## 9.7.5 Member Data Documentation

### 9.7.5.1 `exptype iegenlib::Exp::mExpType [private]`

### 9.7.5.2 `std::list<Term*> iegenlib::Exp::mTerms [private]`

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

- [src/iegenlib/set\\_relation/expression.h](#)
- [src/iegenlib/set\\_relation/expression.cc](#)

## 9.8 ExpTest Class Reference

## Public Attributes

- `std::string` [root\\_dir](#)
- `std::string` [dot\\_data](#)

## Protected Member Functions

- `void` [SetUp](#) ()
- `void` [TearDown](#) ()

### 9.8.1 Member Function Documentation

9.8.1.1 `void` [ExpTest::SetUp](#) ( ) [protected]

9.8.1.2 `void` [ExpTest::TearDown](#) ( ) [inline, protected]

### 9.8.2 Member Data Documentation

9.8.2.1 `std::string` [ExpTest::dot\\_data](#)

9.8.2.2 `std::string` [ExpTest::root\\_dir](#)

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

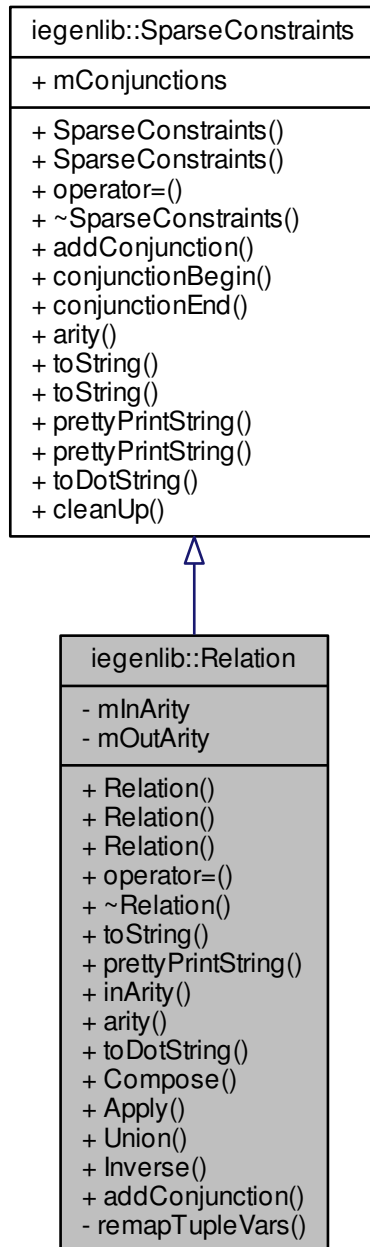
- [src/iegenlib/set\\_relation/expression\\_test.cc](#)

## 9.9 iegenlib::Relation Class Reference

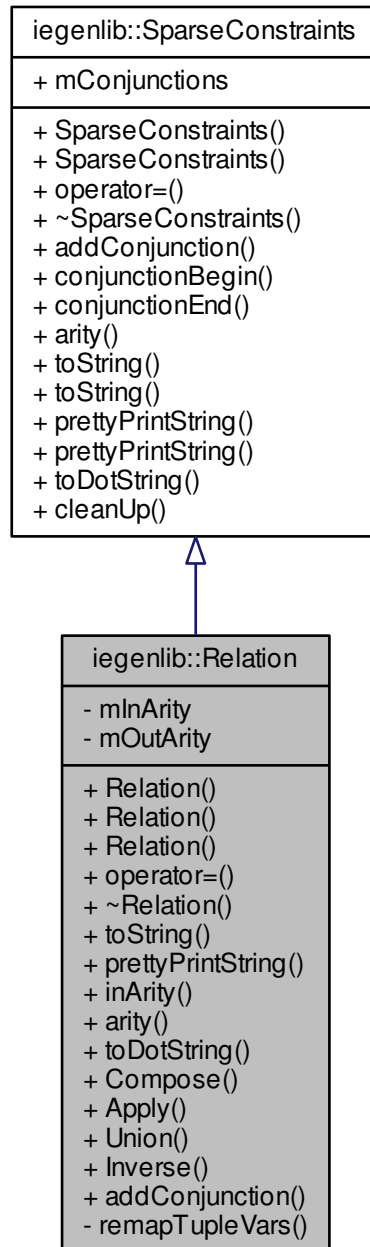
A [SparseConstraints](#) class that represents a [Relation](#).

```
#include <set_relation.h>
```

Inheritance diagram for iegenlib::Relation:



Collaboration diagram for `iegenlib::Relation`:



## Public Member Functions

- [Relation](#) (std::string str)
- [Relation](#) (int inArity, int outArity)
- [Relation](#) (const [Relation](#) &other)
- [Relation](#) & [operator=](#) (const [Relation](#) &other)
- [~Relation](#) ()
- std::string [toString](#) () const  
*Convert to a human-readable string.*
- std::string [prettyPrintString](#) () const  
*Convert to a human-readable string (substitute in tuple vars).*
- int [inArity](#) () const  
*Get our in arity.*
- int [arity](#) () const  
*Get our total arity (in plus out).*
- std::string [toDotString](#) () const
- [Relation](#) \* [Compose](#) (const [Relation](#) \*rhs) const
- [Set](#) \* [Apply](#) (const [Set](#) \*rhs) const
- [Relation](#) \* [Union](#) (const [Relation](#) \*rhs) const
- [Relation](#) \* [Inverse](#) () const
- void [addConjunction](#) ([Conjunction](#) \*adoptedConjunction)

## Private Member Functions

- void [remapTupleVars](#) (const std::vector< int > &oldToNewLocs)

## Private Attributes

- int [mInArity](#)
- int [mOutArity](#)

### 9.9.1 Detailed Description

A [SparseConstraints](#) class that represents a [Relation](#).

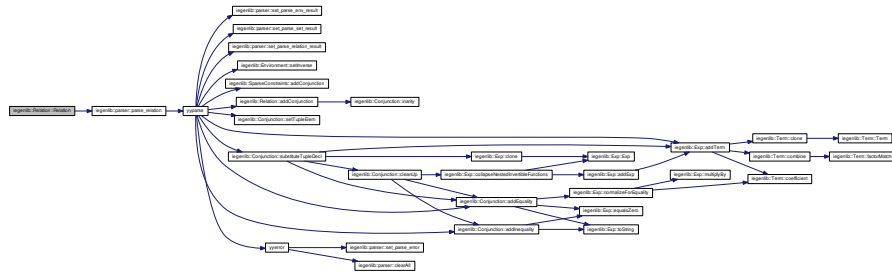
This class has two arities related to it to indicate the in arity and the out arity.

Representation example: {[x]->[y]:x < 100 and y > 0}

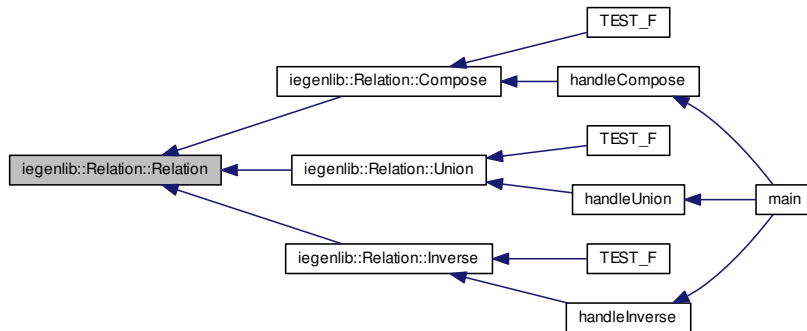
## 9.9.2 Constructor & Destructor Documentation

### 9.9.2.1 iegenlib::Relation::Relation ( std::string str )

Here is the call graph for this function:



Here is the caller graph for this function:



### 9.9.2.2 iegenlib::Relation::Relation ( int inArity, int outArity )

### 9.9.2.3 iegenlib::Relation::Relation ( const Relation & other )

### 9.9.2.4 iegenlib::Relation::~~Relation ( )

## 9.9.3 Member Function Documentation



9.9.3.1 void iegenlib::Relation::addConjunction ( Conjunction \* adoptedConjunction )  
 [inline, virtual]

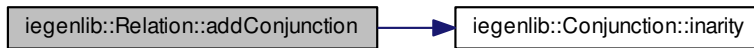
addConjunction that checks the [Conjunction](#) and [Relation](#) arities match

Parameters

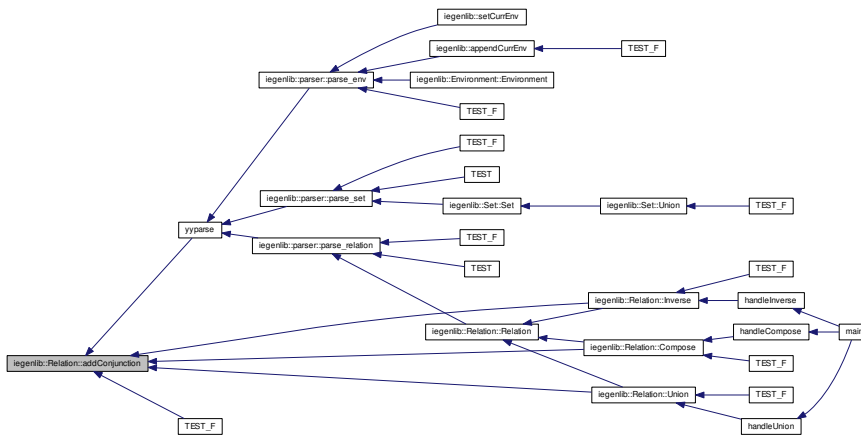
<i>adoptedcon-</i> <i>junction</i>	(adopted)
---------------------------------------	-----------

Reimplemented from [iegenlib::SparseConstraints](#).

Here is the call graph for this function:



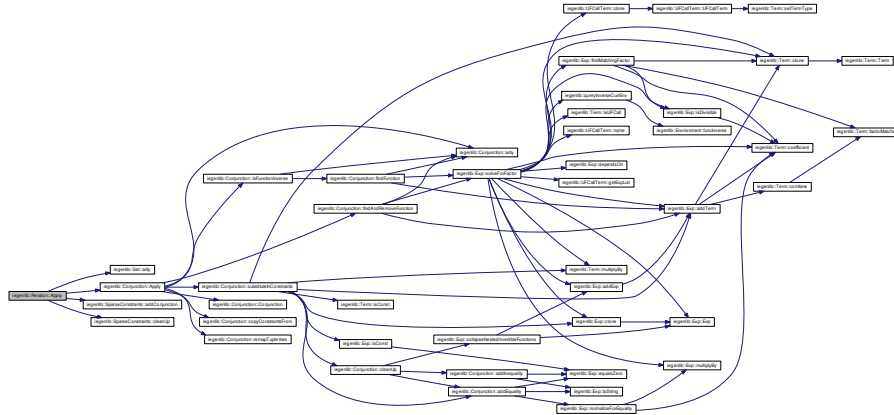
Here is the caller graph for this function:



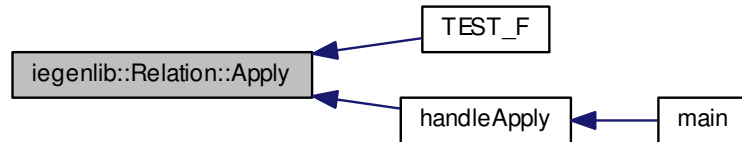
### 9.9.3.2 Set \* iegenlib::Relation::Apply ( const Set \* rhs ) const

Apply this relation to the given set. Returns a new [Set](#), which the caller is responsible for deallocating.

Here is the call graph for this function:



Here is the caller graph for this function:



### 9.9.3.3 int iegenlib::Relation::arity ( ) const [inline, virtual]

Get our total arity (in plus out).

Reimplemented from [iegenlib::SparseConstraints](#).

9.9.3.4 Relation \* iegenlib::Relation::Compose ( const Relation \* rhs ) const

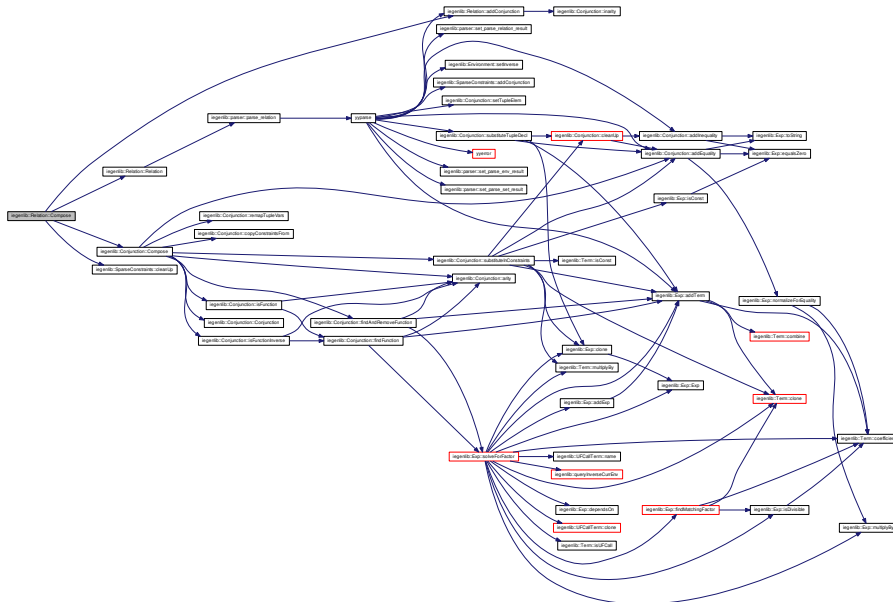
Compose this relation with the given other one (i.e., this Compose rhs). Returns a new Relation, which the caller is responsible for deallocating.

Parameters

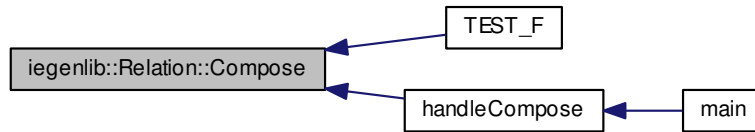
<i>rhs</i>	(not adopted)
------------	---------------

Compose this relation with the given other one (i.e., this Compose rhs). Returns a new Relation, which the caller is responsible for deallocating.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 9.9.3.5 `int iegenlib::Relation::inArity ( ) const` `[inline]`

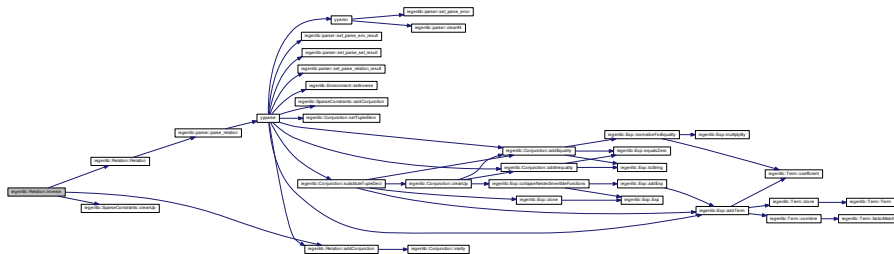
Get our in arity.

#### 9.9.3.6 `Relation * iegenlib::Relation::Inverse ( ) const`

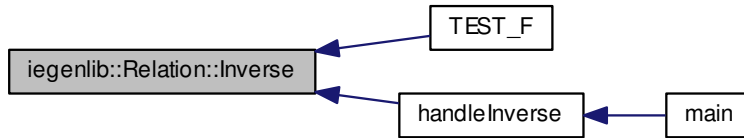
Create the inverse of this relation. Returns a new [Relation](#), which the caller is responsible for deallocating.

Inverse this relation. Returns a new [Relation](#), which the caller is responsible for deallocating.

Here is the call graph for this function:



Here is the caller graph for this function:



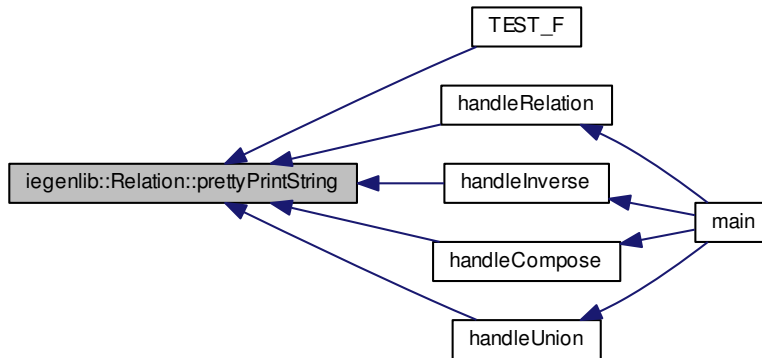
#### 9.9.3.7 Relation & iegenlib::Relation::operator= ( const Relation & other )

#### 9.9.3.8 std::string iegenlib::Relation::prettyPrintString ( )const [virtual]

Convert to a human-readable string (substitute in tuple vars).

Reimplemented from [iegenlib::SparseConstraints](#).

Here is the caller graph for this function:



9.9.3.9 void `iegenlib::Relation::remapTupleVars` ( const `std::vector< int >` & `oldToNewLocs` )  
 [private]

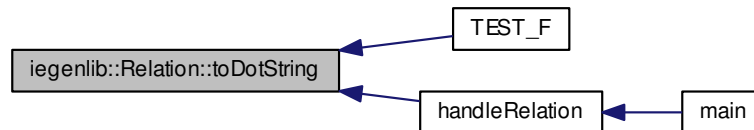
Find any `TupleVarTerms` in this expression (and subexpressions) and remap the locations according to the `oldToNewLocs` vector, where `oldToNewLocs[i] = j` means that old location `i` becomes new location `j` (i.e. `__tvi -> __tvj`). Throws an exception if an old location is out of range for the given `oldToNewLocs`.

9.9.3.10 `std::string` `iegenlib::Relation::toDotString` ( ) const [virtual]

Create a string for use with the GraphViz tool `dot` so we can visualize the [Relation](#) data structure.

Reimplemented from [iegenlib::SparseConstraints](#).

Here is the caller graph for this function:

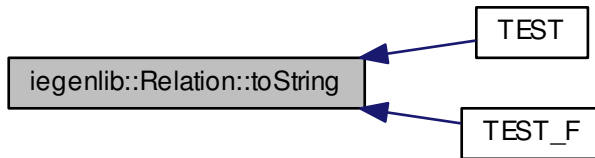


9.9.3.11 `std::string` `iegenlib::Relation::toString` ( ) const [virtual]

Convert to a human-readable string.

Reimplemented from [iegenlib::SparseConstraints](#).

Here is the caller graph for this function:



9.9.3.12 Relation \* iegenlib::Relation::Union ( const Relation \* rhs ) const

Union this relation with the given other one (i.e., the Union rhs). Returns a new Relation, which the caller is responsible for deallocating.

Parameters

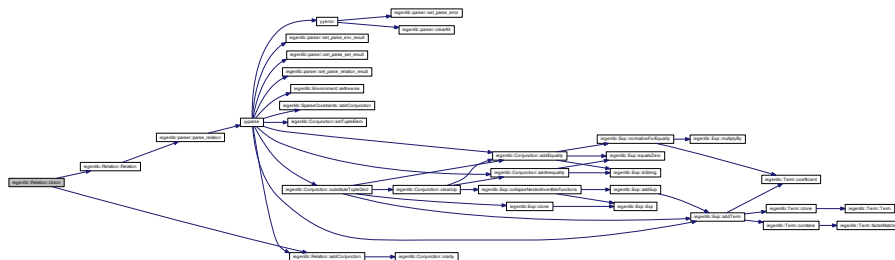
<i>rhs</i>	(not adopted)
------------	---------------

Union this relation with the given other one (i.e., this Union rhs). Returns a new Relation, which the caller is responsible for deallocating.

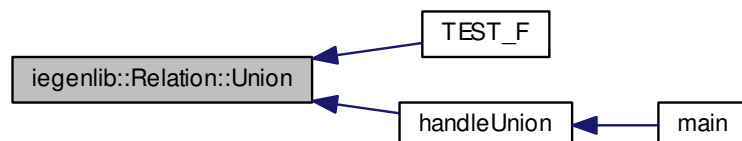
Parameters

<i>rhs</i>	(not adopted)
------------	---------------

Here is the call graph for this function:



Here is the caller graph for this function:



#### 9.9.4 Member Data Documentation

9.9.4.1 `int iegenlib::Relation::mInArity` [private]

9.9.4.2 `int iegenlib::Relation::mOutArity` [private]

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

- [src/iegenlib/set\\_relation/set\\_relation.h](#)
- [src/iegenlib/set\\_relation/set\\_relation.cc](#)

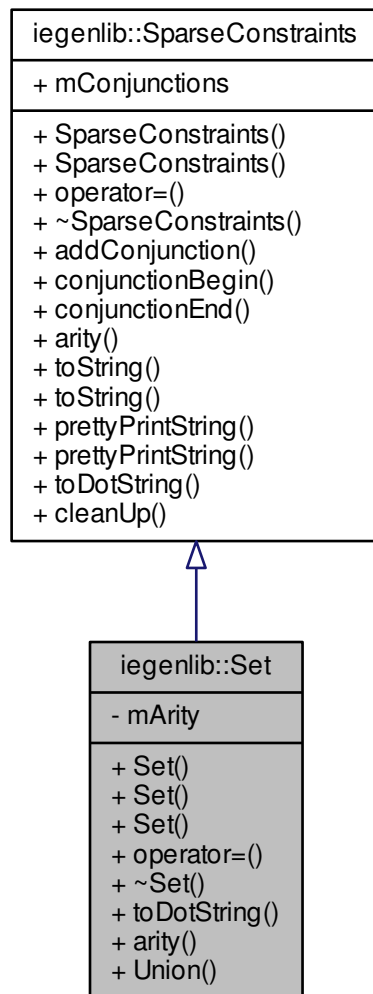
### 9.10 iegenlib::Set Class Reference

A [SparseConstraints](#) class that represents a [Set](#).

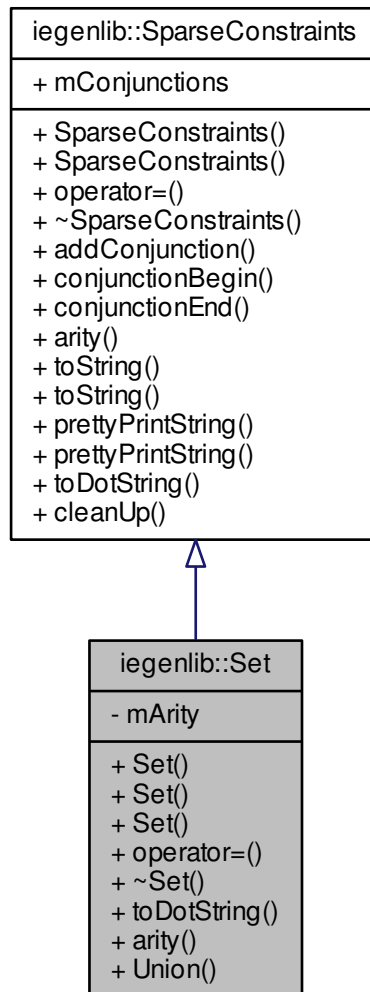
```
#include <set_relation.h>
```



Inheritance diagram for iegenlib::Set:



Collaboration diagram for `iegenlib::Set`:



### Public Member Functions

- [Set](#) (`std::string str`)

- [Set](#) (int arity)
- [Set](#) (const [Set](#) &other)
- [Set](#) & [operator=](#) (const [Set](#) &other)
- [~Set](#) ()
- `std::string toDotString ()` const  
Create a graph for visualization with `graphviz`.
- `int arity ()` const  
Get our total arity, IOW number of tuple elements.
- [Set](#) \* [Union](#) (const [Set](#) \*rhs) const

### Private Attributes

- `int mArity`

#### 9.10.1 Detailed Description

A [SparseConstraints](#) class that represents a [Set](#).

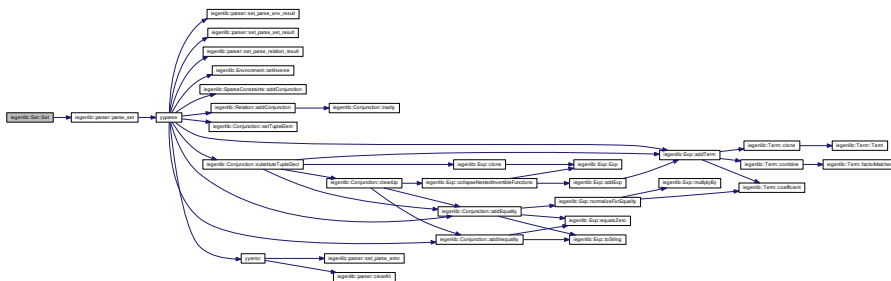
This class has one arity related to it.

Representation example: `{[x]:x < 100}`

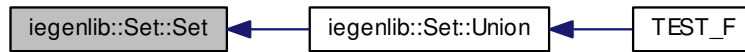
#### 9.10.2 Constructor & Destructor Documentation

##### 9.10.2.1 `iegenlib::Set::Set ( std::string str )`

Here is the call graph for this function:



Here is the caller graph for this function:



9.10.2.2 `iegenlib::Set::Set ( int arity )`

9.10.2.3 `iegenlib::Set::Set ( const Set & other )`

9.10.2.4 `iegenlib::Set::~~Set ( )`

### 9.10.3 Member Function Documentation

9.10.3.1 `int iegenlib::Set::arity ( ) const [inline, virtual]`

Get our total arity, IOW number of tuple elements.

Reimplemented from [iegenlib::SparseConstraints](#).

Here is the caller graph for this function:



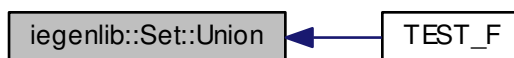
9.10.3.2 `Set & iegenlib::Set::operator= ( const Set & other )`

9.10.3.3 `std::string iegenlib::Set::toDotString ( ) const [virtual]`

Create a graph for visualization with graphviz.



Here is the caller graph for this function:



## 9.10.4 Member Data Documentation

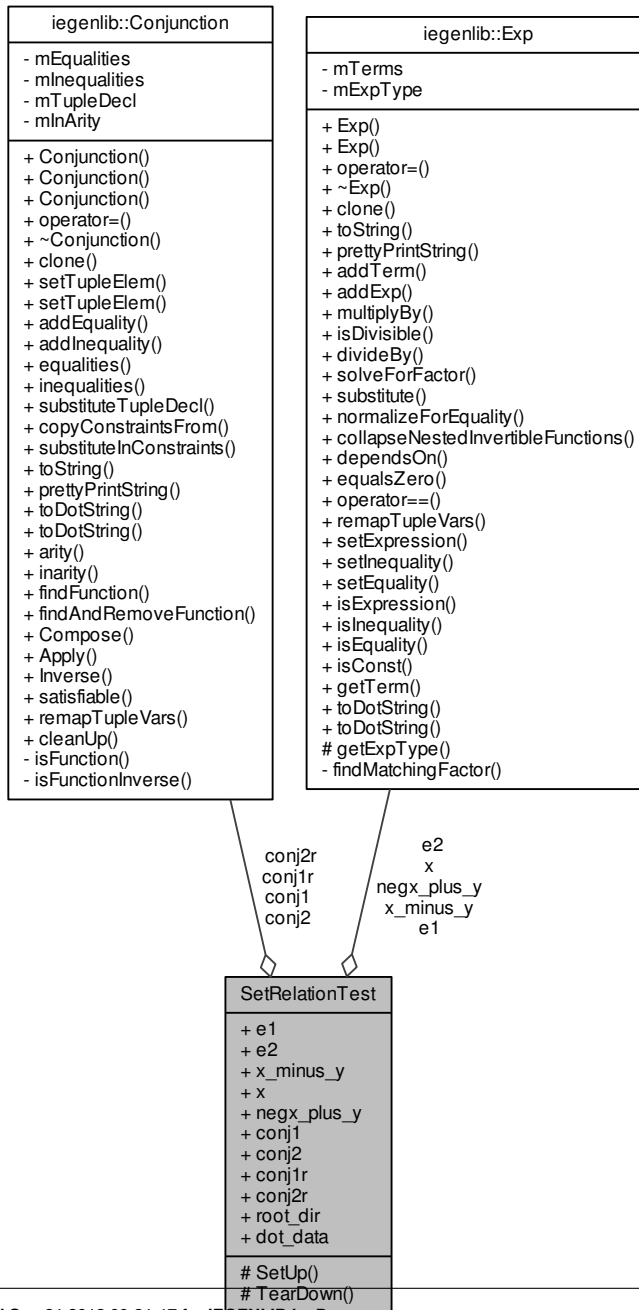
9.10.4.1 `int iegenlib::Set::mArity` [private]

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

- [src/iegenlib/set\\_relation/set\\_relation.h](#)
- [src/iegenlib/set\\_relation/set\\_relation.cc](#)

## 9.11 SetRelationTest Class Reference

Collaboration diagram for SetRelationTest:



## Public Attributes

- [Exp](#) e1
- [Exp](#) e2
- [Exp](#) x\_minus\_y
- [Exp](#) x
- [Exp](#) negx\_plus\_y
- [Conjunction](#) \* conj1
- [Conjunction](#) \* conj2
- [Conjunction](#) \* conj1r
- [Conjunction](#) \* conj2r
- `std::string` root\_dir
- `std::string` dot\_data

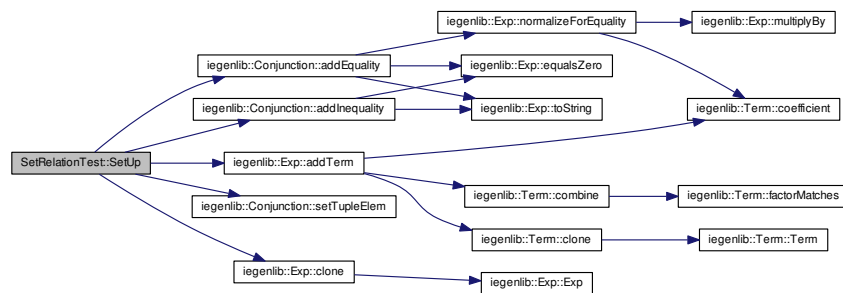
## Protected Member Functions

- virtual void [SetUp](#) ()
- virtual void [TearDown](#) ()

### 9.11.1 Member Function Documentation

#### 9.11.1.1 void SetRelationTest::SetUp ( ) [protected, virtual]

Here is the call graph for this function:





9.11.1.2 void SetRelationTest::TearDown ( ) [protected, virtual]

## 9.11.2 Member Data Documentation

9.11.2.1 Conjunction\* SetRelationTest::conj1

9.11.2.2 Conjunction\* SetRelationTest::conj1r

9.11.2.3 Conjunction\* SetRelationTest::conj2

9.11.2.4 Conjunction\* SetRelationTest::conj2r

9.11.2.5 std::string SetRelationTest::dot\_data

9.11.2.6 Exp SetRelationTest::e1

9.11.2.7 Exp SetRelationTest::e2

9.11.2.8 Exp SetRelationTest::negx\_plus\_y

9.11.2.9 std::string SetRelationTest::root\_dir

9.11.2.10 Exp SetRelationTest::x

9.11.2.11 Exp SetRelationTest::x\_minus\_y

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

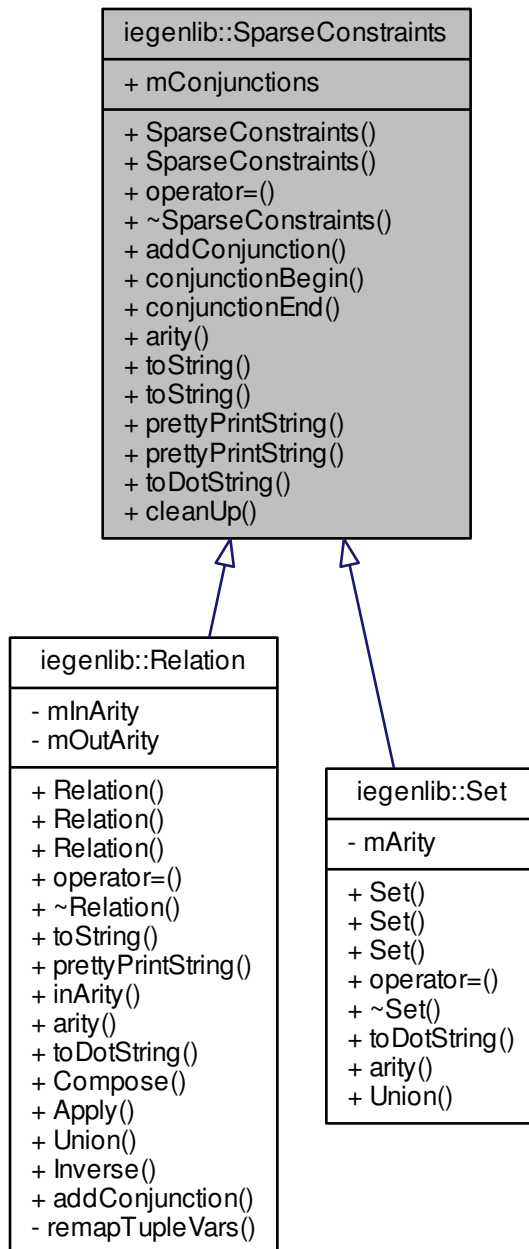
- src/iegenlib/set\_relation/[set\\_relation\\_test.cc](#)

## 9.12 iegenlib::SparseConstraints Class Reference

Base class that contains the conjunctions and a pointer to an environment.

```
#include <set_relation.h>
```

Inheritance diagram for iegenlib::SparseConstraints:



## Public Member Functions

- [SparseConstraints](#) ()
- [SparseConstraints](#) (const [SparseConstraints](#) &other)
- [SparseConstraints](#) & operator= (const [SparseConstraints](#) &other)
- virtual [~SparseConstraints](#) ()
- virtual void [addConjunction](#) ([Conjunction](#) \*adoptedConjunction)
- std::list< [Conjunction](#) \* > ::const\_iterator [conjunctionBegin](#) () const  
*Get an iterator to the first conjunction we contain.*
- std::list< [Conjunction](#) \* > ::const\_iterator [conjunctionEnd](#) () const  
*Get an iterator pointing past the last conjunction we contain.*
- virtual int [arity](#) () const
- virtual std::string [toString](#) () const
- std::string [toString](#) (int aritySplit) const
- virtual std::string [prettyPrintString](#) () const  
*Convert to a human-readable string (substitute in tuple vars).*
- std::string [prettyPrintString](#) (int aritySplit) const
- virtual std::string [toDotString](#) () const  
*Create a graph for visualization with graphviz.*
- void [cleanUp](#) ()  
*Remove duplicate constraints and trivial constraints.*

## Public Attributes

- std::list< [Conjunction](#) \* > [mConjunctions](#)

### 9.12.1 Detailed Description

Base class that contains the conjunctions and a pointer to an environment.

The purpose of this class, at the moment, is mostly as a base class and to be extended by set and relation, both of which will have their own special behaviors dealing with the arity.

### 9.12.2 Constructor & Destructor Documentation

9.12.2.1 [iegenlib::SparseConstraints::SparseConstraints](#) ( )

9.12.2.2 [iegenlib::SparseConstraints::SparseConstraints](#) ( const [SparseConstraints](#) & *other* )

9.12.2.3 `iegenlib::SparseConstraints::~~SparseConstraints ( )` [virtual]

### 9.12.3 Member Function Documentation

9.12.3.1 `virtual void iegenlib::SparseConstraints::addConjunction ( Conjunction * adoptedConjunction )` [inline, virtual]

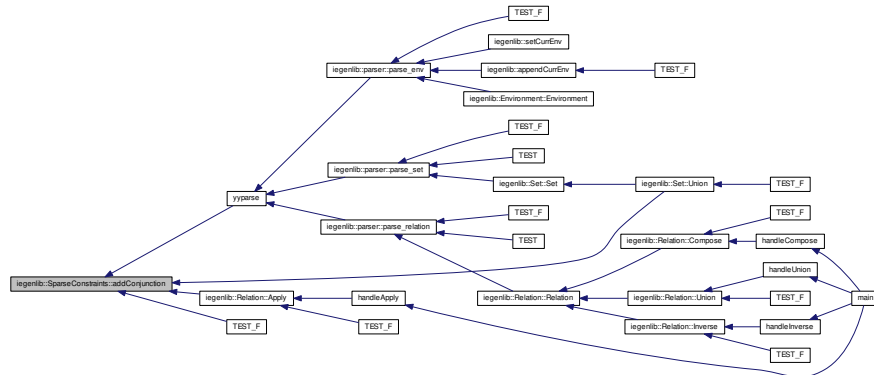
addConjunction

#### Parameters

<i>adoptedcon- junction</i>	(adopted)
---------------------------------	-----------

Reimplemented in [iegenlib::Relation](#).

Here is the caller graph for this function:

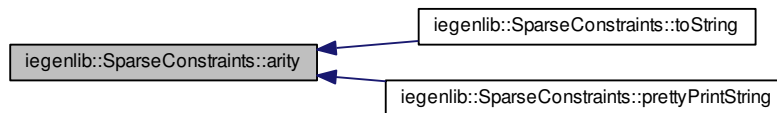


9.12.3.2 `virtual int iegenlib::SparseConstraints::arity ( ) const` [inline, virtual]

Get our total arity, IOW number of tuple elements. Should be overridden in subclasses.

Reimplemented in [iegenlib::Relation](#), and [iegenlib::Set](#).

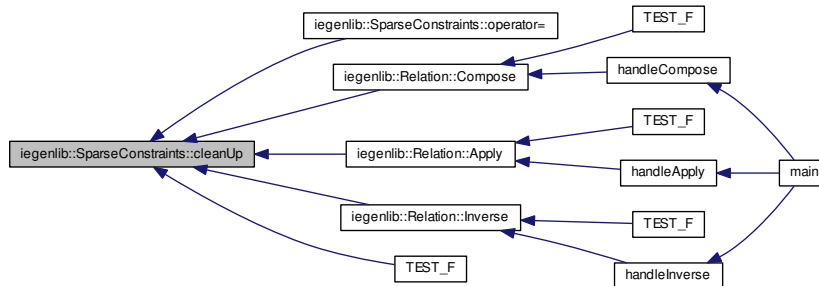
Here is the caller graph for this function:



### 9.12.3.3 void iegenlib::SparseConstraints::cleanUp ( )

Remove duplicate constraints and trivial constraints.

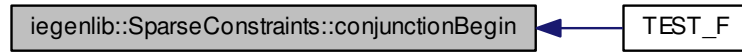
Here is the caller graph for this function:



### 9.12.3.4 std::list<Conjunction\*>::const\_iterator iegenlib::SparseConstraints::conjunction- Begin ( ) const [inline]

Get an iterator to the first conjunction we contain.

Here is the caller graph for this function:



9.12.3.5 `std::list<Conjunction*>::const_iterator iegenlib::SparseConstraints::conjunction-End ( ) const [inline]`

Get an iterator pointing past the last conjunction we contain.

9.12.3.6 `SparseConstraints & iegenlib::SparseConstraints::operator= ( const SparseConstraints & other )`

Here is the call graph for this function:

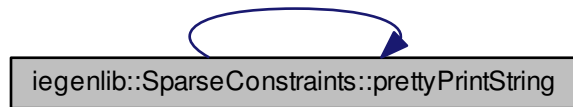


9.12.3.7 `virtual std::string iegenlib::SparseConstraints::prettyPrintString ( ) const [inline, virtual]`

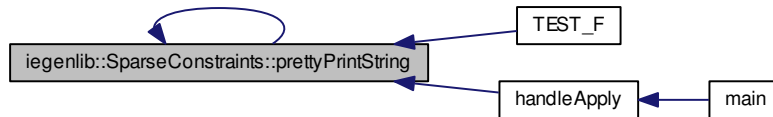
Convert to a human-readable string (substitute in tuple vars).

Reimplemented in [iegenlib::Relation](#).

Here is the call graph for this function:

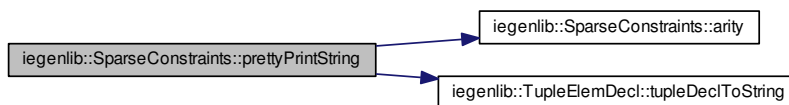


Here is the caller graph for this function:



#### 9.12.3.8 `std::string iegenlib::SparseConstraints::prettyPrintString ( int aritySplit ) const`

Here is the call graph for this function:



### 9.12.3.9 `std::string iegenlib::SparseConstraints::toDotString ( ) const` [virtual]

Create a graph for visualization with graphviz.

Reimplemented in [iegenlib::Relation](#), and [iegenlib::Set](#).

Here is the caller graph for this function:

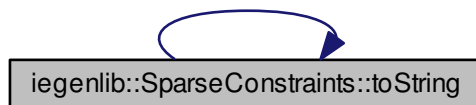


### 9.12.3.10 `virtual std::string iegenlib::SparseConstraints::toString ( ) const` [inline, virtual]

Convert to a human-readable string. Still need arity split here because works for Sets and Relations.

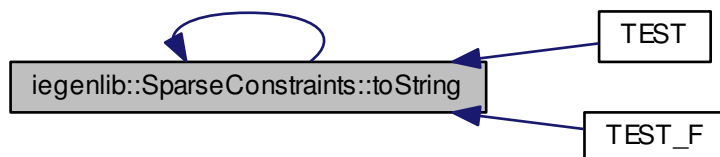
Reimplemented in [iegenlib::Relation](#).

Here is the call graph for this function:



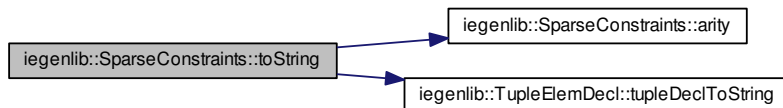


Here is the caller graph for this function:



9.12.3.11 `std::string iegenlib::SparseConstraints::toString ( int aritySplit ) const`

Here is the call graph for this function:



## 9.12.4 Member Data Documentation

9.12.4.1 `std::list<Conjunction*> iegenlib::SparseConstraints::mConjunctions`

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

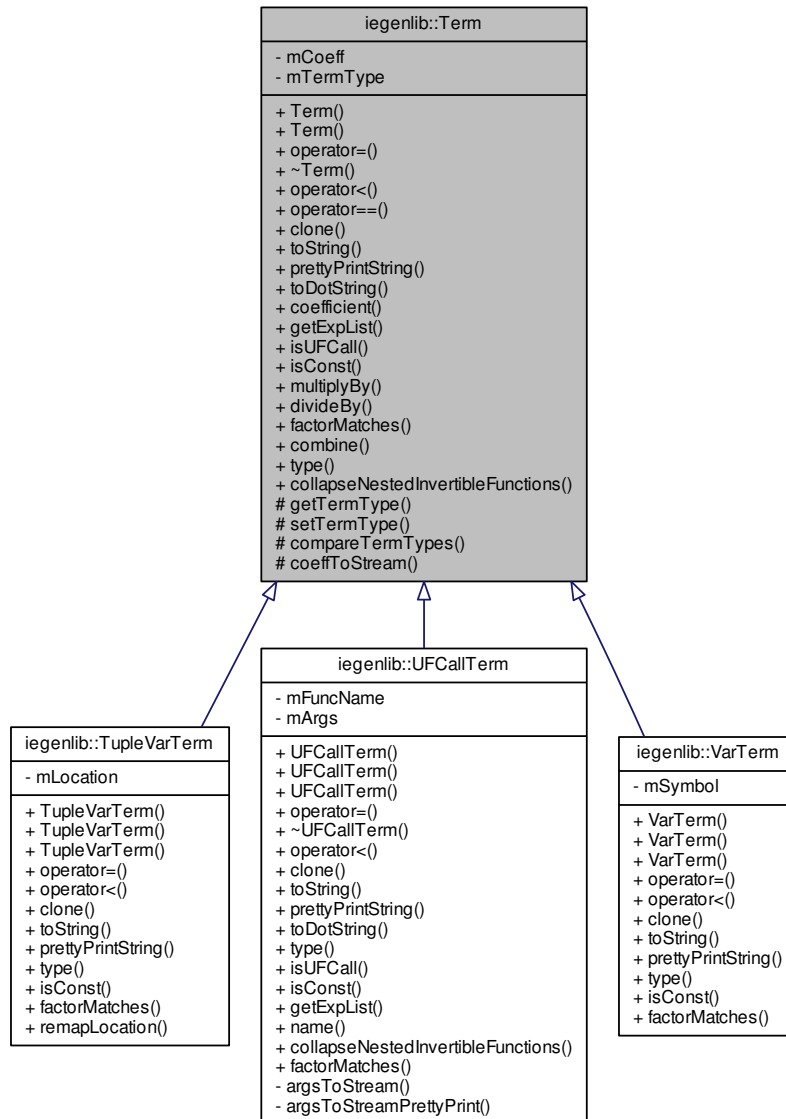
- [src/iegenlib/set\\_relation/set\\_relation.h](#)
- [src/iegenlib/set\\_relation/set\\_relation.cc](#)

## 9.13 iegenlib::Term Class Reference

A coefficient multiplied by one. Subclasses are multiplied by other entities.

```
#include <expression.h>
```

Inheritance diagram for `iegenlib::Term`:



## Public Member Functions

- [Term](#) (int coeff)  
*Default constructor.*
- [Term](#) (const [Term](#) &other)  
*Copy constructor.*
- virtual [Term](#) & [operator=](#) (const [Term](#) &other)  
*Copy assignment.*
- virtual [~Term](#) ()  
*Destructor.*
- virtual bool [operator<](#) (const [Term](#) &other) const  
*Comparison operator -- lexicographic order.*
- bool [operator==](#) (const [Term](#) &other) const  
*equality operator*
- virtual [Term](#) \* [clone](#) () const  
*Create a copy of this [Term](#) (and of the same subclass)*
- virtual std::string [toString](#) (bool absValue=false) const  
*Creates a compact string to help with debugging.*
- virtual std::string [prettyPrintString](#) (const std::vector< [TupleElemDecl](#) > &m-TupleDecl, bool absValue=false) const  
*Creates a compact string, pretty printed.*
- virtual std::string [toDotString](#) (bool absValue=false) const  
*Creates a brief compact string to help with DOT output.*
- int [coefficient](#) () const  
*Get the coefficient of this term.*
- virtual bool [getExpList](#) (std::list< [Exp](#) \* > \*\*outList)  
*Get a list of any expressions contained by this term.*
- virtual bool [isUFCall](#) () const  
*Returns true if the [Term](#) is really a [UFCallTerm](#).*
- virtual bool [isConst](#) () const  
*Returns true if the [Term](#) is a const.*
- void [multiplyBy](#) (int constant)  
*Multiply the coefficient by a constant.*
- void [divideBy](#) (int divisor)  
*Divide the coefficient by a constant.*
- virtual bool [factorMatches](#) (const [Term](#) &other) const  
*Returns true if this term can be combined with the given term.*
- virtual bool [combine](#) ([Term](#) \*other)
- virtual std::string [type](#) () const  
*Returns string of subclass type.*
- virtual [Exp](#) \* [collapseNestedInvertibleFunctions](#) () const

## Protected Types

- enum `termtype` { `TupleVar`, `SymConst`, `UFCall`, `ConstVal` }

## Protected Member Functions

- `termtype` `getTermType` () const
- void `setTermType` (`termtype` tt)
- int `compareTermTypes` (const `Term` &other) const
- void `coeffToStream` (std::stringstream &ss, bool absValue) const

## Private Attributes

- int `mCoeff`
- `termtype` `mTermType`

### 9.13.1 Detailed Description

A coefficient multiplied by one. Subclasses are multiplied by other entities.

### 9.13.2 Member Enumeration Documentation

9.13.2.1 enum `iegenlib::Term::termtype` [protected]

Enumerator:

***TupleVar***

***SymConst***

***UFCall***

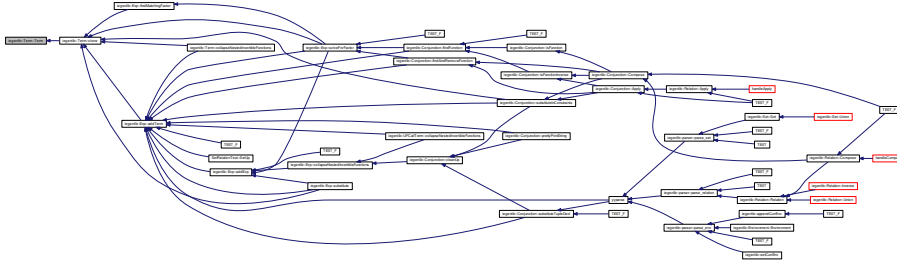
***ConstVal***

### 9.13.3 Constructor & Destructor Documentation

9.13.3.1 `iegenlib::Term::Term` ( int *coeff* ) [inline]

Default constructor.

Here is the caller graph for this function:



### 9.13.3.2 iegenlib::Term::Term ( const Term & other )

Copy constructor.

### 9.13.3.3 virtual iegenlib::Term::~~Term ( ) [inline, virtual]

Destructor.

## 9.13.4 Member Function Documentation

### 9.13.4.1 Term \* iegenlib::Term::clone ( ) const [virtual]

Create a copy of this [Term](#) (and of the same subclass)

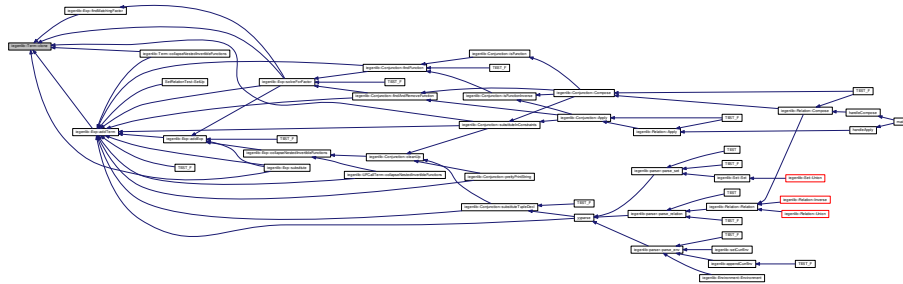
We need a clone method because [Term](#) has subclasses. We don't always know what copy constructor to use.

Reimplemented in [iegenlib::VarTerm](#), [iegenlib::TupleVarTerm](#), and [iegenlib::UFCallTerm](#).

Here is the call graph for this function:



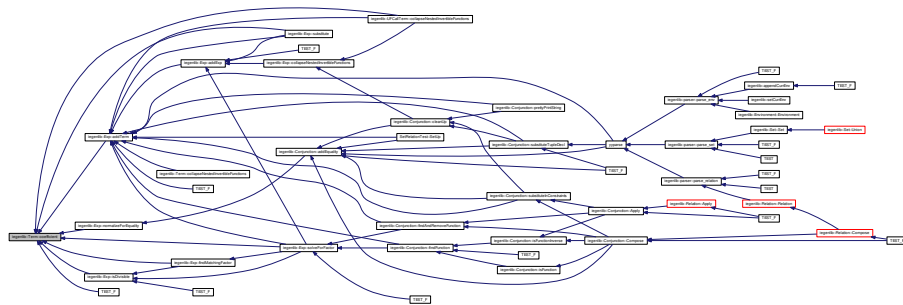
Here is the caller graph for this function:



#### 9.13.4.2 `int iegenlib::Term::coefficient ( ) const` `[inline]`

Get the coefficient of this term.

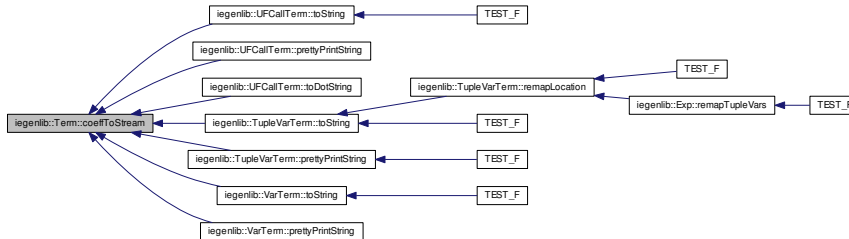
Here is the caller graph for this function:



#### 9.13.4.3 `void iegenlib::Term::coeffToStream ( std::stringstream & ss, bool absValue ) const` `[protected]`

`coeffToStream`: output the coefficient to the stream, if it is not equal to one; this is used mainly by the subclasses, as part of their `toString()` process.

Here is the caller graph for this function:

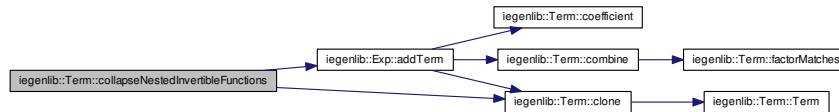


#### 9.13.4.4 `Exp * iegenlib::Term::collapseNestedInvertibleFunctions ( ) const` [virtual]

Return a new `Exp` with all nested functions such as `f ( f_inv ( i ) )` changed to `i`.

Reimplemented in [iegenlib::UFCallTerm](#).

Here is the call graph for this function:



#### 9.13.4.5 `bool iegenlib::Term::combine ( Term * other )` [virtual]

Combine another term into this one, if possible, by adding coefficients of corresponding factors.

##### Parameters

<code>other</code>	-- term to attempt to combine with this one.(adopt)
--------------------	---

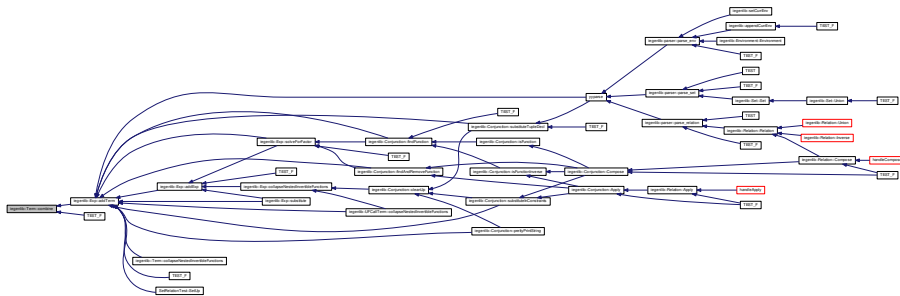
**Returns**

true if other was combined with this one; false otherwise

Here is the call graph for this function:



Here is the caller graph for this function:



#### 9.13.4.6 `int iegenlib::Term::compareTermTypes ( const Term & other ) const` [protected]

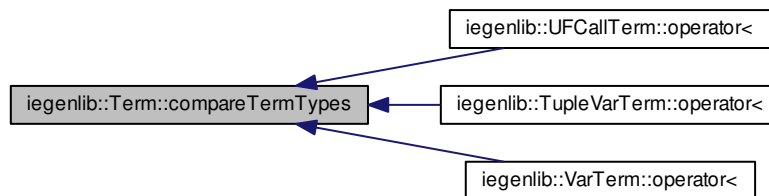
Return -1 if this `mTermType < other.mTermType`, 1 if this `mTermType > other.mTermType`, and 0 if the `mTermTypes` are identical.

**Parameters**

<i>other, object</i>	to be compared
----------------------	----------------



Here is the caller graph for this function:



#### 9.13.4.7 void iegenlib::Term::divideBy ( int *divisor* )

Divide the coefficient by a constant.

Here is the caller graph for this function:



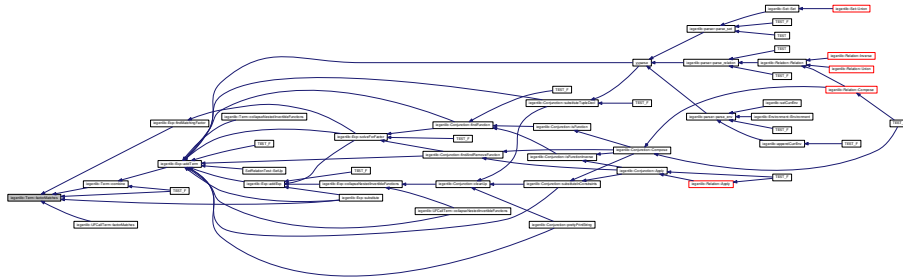
#### 9.13.4.8 bool iegenlib::Term::factorMatches ( const Term & *other* ) const [virtual]

Returns true if this term can be combined with the given term.

Returns true if this term has the same factor (i.e. everything except the coefficient) as the given other term.

Reimplemented in [iegenlib::VarTerm](#), [iegenlib::TupleVarTerm](#), and [iegenlib::UFCallTerm](#).

Here is the caller graph for this function:



**9.13.4.9** `virtual bool iegenlib::Term::getExpList ( std::list< Exp * > ** outList )`  
`[inline, virtual]`

Get a list of any expressions contained by this term.

Reimplemented in [iegenlib::UFCallTerm](#).

Here is the caller graph for this function:



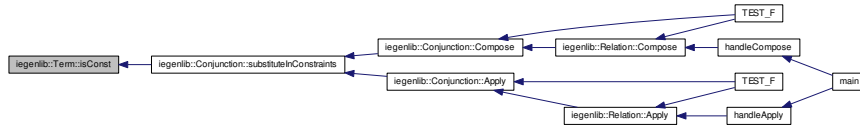
**9.13.4.10** `termtype iegenlib::Term::getTermType ( ) const` `[inline, protected]`

**9.13.4.11** `virtual bool iegenlib::Term::isConst ( ) const` `[inline, virtual]`

Returns true if the [Term](#) is a const.

Reimplemented in [iegenlib::VarTerm](#), [iegenlib::TupleVarTerm](#), and [iegenlib::UFCallTerm](#).

Here is the caller graph for this function:



9.13.4.12 virtual bool iegenlib::Term::isUFCall ( ) const [inline, virtual]

Returns true if the Term is really a [UFCallTerm](#).

Reimplemented in [iegenlib::UFCallTerm](#).

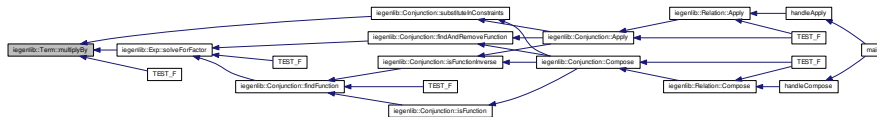
Here is the caller graph for this function:



9.13.4.13 void iegenlib::Term::multiplyBy ( int constant )

Multiply the coefficient by a constant.

Here is the caller graph for this function:



9.13.4.14 bool iegenlib::Term::operator< ( const Term & other ) const [virtual]

Comparison operator -- lexicographic order.

Compare two terms in a canonical order, defined as follows: 1. by term type: TupleVar, SymConst, UFCall, ConstVal 2. within each type, in alphabetical or numeric order.

#### Parameters

<i>other, object</i>	to be compared
----------------------	----------------

Reimplemented in [iegenlib::VarTerm](#), [iegenlib::TupleVarTerm](#), and [iegenlib::UFCallTerm](#).

#### 9.13.4.15 Term & iegenlib::Term::operator= ( const Term & other ) [virtual]

Copy assignment.

#### 9.13.4.16 bool iegenlib::Term::operator== ( const Term & other ) const [inline]

equality operator

#### 9.13.4.17 std::string iegenlib::Term::prettyPrintString ( const std::vector< TupleElemDecl > & mTupleDecl, bool absValue = false ) const [virtual]

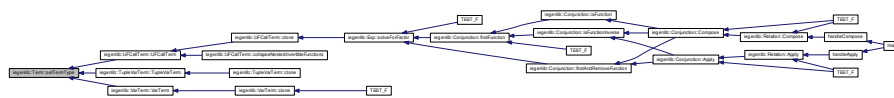
Creates a compact string, pretty printed.

Creates a compact string pretty printed.

Reimplemented in [iegenlib::VarTerm](#), [iegenlib::TupleVarTerm](#), and [iegenlib::UFCallTerm](#).

#### 9.13.4.18 void iegenlib::Term::setTermType ( termtype tt ) [inline, protected]

Here is the caller graph for this function:



#### 9.13.4.19 std::string iegenlib::Term::toDotString ( bool absValue = false ) const [virtual]

Creates a brief compact string to help with DOT output.

Reimplemented in [iegenlib::UFCallTerm](#).

Here is the call graph for this function:

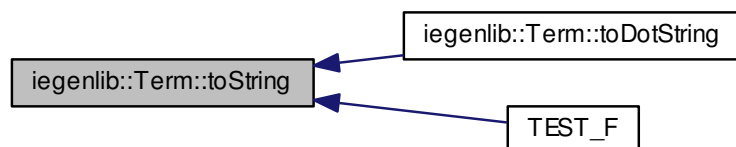


9.13.4.20 `std::string iegenlib::Term::toString ( bool absValue = false ) const`  
[virtual]

Creates a compact string to help with debugging.

Reimplemented in [iegenlib::VarTerm](#), [iegenlib::TupleVarTerm](#), and [iegenlib::UFCallTerm](#).

Here is the caller graph for this function:



9.13.4.21 `std::string iegenlib::Term::type ( ) const` [virtual]

Returns string of subclass type.

Reimplemented in [iegenlib::VarTerm](#), [iegenlib::TupleVarTerm](#), and [iegenlib::UFCallTerm](#).

### 9.13.5 Member Data Documentation

9.13.5.1 `int iegenlib::Term::mCoeff` [`private`]

9.13.5.2 `termtype iegenlib::Term::mTermType` [`private`]

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

- [src/iegenlib/set\\_relation/expression.h](#)
- [src/iegenlib/set\\_relation/expression.cc](#)

## 9.14 iegenlib::TupleElemDecl Class Reference

Info about a tuple element, which is a constant or a variable.

```
#include <set_relation.h>
```

### Public Member Functions

- [TupleElemDecl](#) ()
- [TupleElemDecl](#) (int constVal)
- [TupleElemDecl](#) (std::string s)
- [TupleElemDecl](#) (const [TupleElemDecl](#) &other)
- [TupleElemDecl](#) & [operator=](#) (const [TupleElemDecl](#) &rhs)
- std::string [toString](#) () const
- bool [isConst](#) () const
- int [constVal](#) () const
- std::string [varString](#) () const
- [Term](#) \* [createTerm](#) (int loc) const

### Static Public Member Functions

- static std::string [tupleDeclToString](#) (const std::vector< [TupleElemDecl](#) > &m-TupleDecl, unsigned int aritySplit=0)

### Private Attributes

- bool [mIsConst](#)
- int [mConstVal](#)
- std::string [mVarString](#)

### 9.14.1 Detailed Description

Info about a tuple element, which is a constant or a variable.

### 9.14.2 Constructor & Destructor Documentation

9.14.2.1 `iegenlib::TupleElemDecl::TupleElemDecl ( )` `[inline]`

9.14.2.2 `iegenlib::TupleElemDecl::TupleElemDecl ( int constVal )` `[inline]`

9.14.2.3 `iegenlib::TupleElemDecl::TupleElemDecl ( std::string s )` `[inline]`

9.14.2.4 `iegenlib::TupleElemDecl::TupleElemDecl ( const TupleElemDecl & other )`  
`[inline]`

### 9.14.3 Member Function Documentation

9.14.3.1 `int iegenlib::TupleElemDecl::constVal ( ) const` `[inline]`

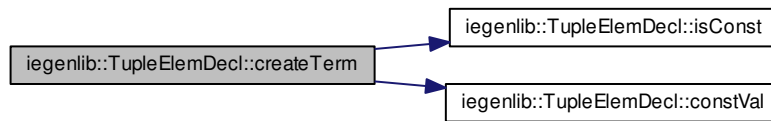
Here is the caller graph for this function:



9.14.3.2 `Term * iegenlib::TupleElemDecl::createTerm ( int loc ) const`

Returns an appropriate term for this [TupleElemDecl](#). Must pass in location of [TupleElemDecl](#). Caller is responsible for deleting the resulting [Term](#).

Here is the call graph for this function:



9.14.3.3 `bool iegenlib::TupleElemDecl::isConst ( ) const [inline]`

Here is the caller graph for this function:



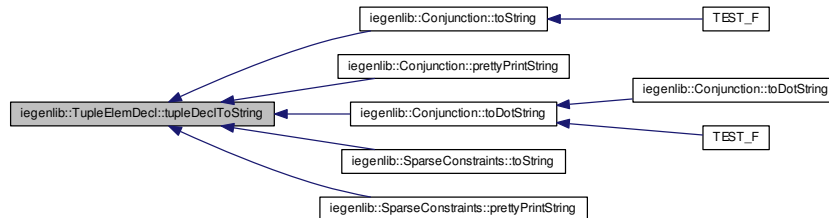
9.14.3.4 `TupleElemDecl& iegenlib::TupleElemDecl::operator=( const TupleElemDecl & rhs ) [inline]`

9.14.3.5 `std::string iegenlib::TupleElemDecl::toString ( ) const`



9.14.3.6 `std::string iegenlib::TupleElemDecl::tupleDeclToString ( const std::vector< TupleElemDecl > & mTupleDecl, unsigned int aritySplit = 0 ) [static]`

Here is the caller graph for this function:



9.14.3.7 `std::string iegenlib::TupleElemDecl::varString ( ) const [inline]`

## 9.14.4 Member Data Documentation

9.14.4.1 `int iegenlib::TupleElemDecl::mConstVal [private]`

9.14.4.2 `bool iegenlib::TupleElemDecl::mIsConst [private]`

9.14.4.3 `std::string iegenlib::TupleElemDecl::mVarString [private]`

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

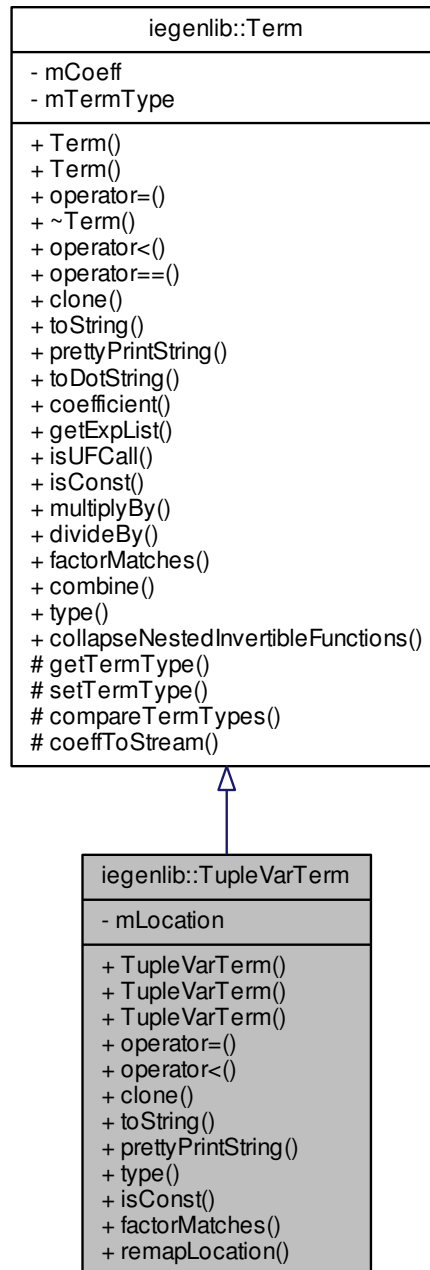
- `src/iegenlib/set_relation/set_relation.h`
- `src/iegenlib/set_relation/set_relation.cc`

## 9.15 iegenlib::TupleVarTerm Class Reference

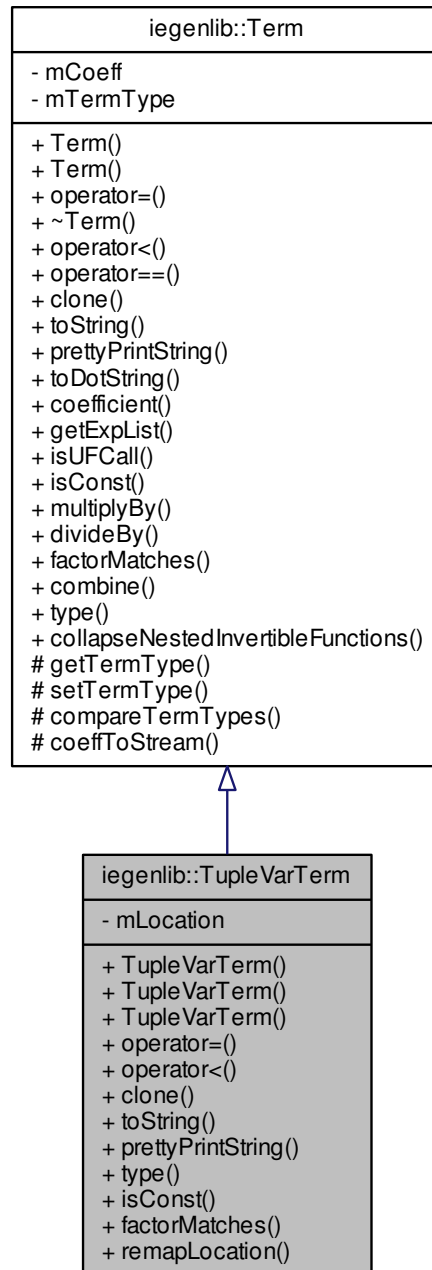
Represents a coefficient multiplied by a tuple variable.

```
#include <expression.h>
```

Inheritance diagram for `iegenlib::TupleVarTerm`:



Collaboration diagram for iegenlib::TupleVarTerm:



## Public Member Functions

- [TupleVarTerm](#) (int coeff, int location)  
*Constructor.*
- [TupleVarTerm](#) (int location)  
*Convenience constructor, assumes coeff = 1.*
- [TupleVarTerm](#) (const [TupleVarTerm](#) &other)  
*Copy constructor.*
- [TupleVarTerm](#) & [operator=](#) (const [TupleVarTerm](#) &other)  
*Copy assignment.*
- bool [operator<](#) (const [Term](#) &other) const  
*Comparison operator -- lexicographic order.*
- [Term](#) \* [clone](#) () const  
*Create a copy of this [Term](#) (and of the same subclass)*
- std::string [toString](#) (bool absValue=false) const  
*Creates a compact string to help with debugging.*
- std::string [prettyPrintString](#) (const std::vector< [TupleElemDecl](#) > &mTupleDecl, bool absValue=false) const  
*Creates a compact string, pretty printed.*
- std::string [type](#) () const  
*Returns string of subclass type.*
- bool [isConst](#) () const  
*Returns true if the [Term](#) is a const.*
- bool [factorMatches](#) (const [Term](#) &other) const  
*Returns true if this term can be combined with the given term.*
- void [remapLocation](#) (const std::vector< int > &oldToNewLocs)

## Private Attributes

- int [mLocation](#)

### 9.15.1 Detailed Description

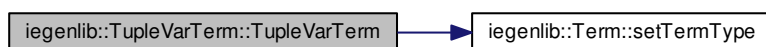
Represents a coefficient multiplied by a tuple variable.

## 9.15.2 Constructor & Destructor Documentation

### 9.15.2.1 iegenlib::TupleVarTerm::TupleVarTerm ( int *coeff*, int *location* ) [inline]

Constructor.

Here is the call graph for this function:



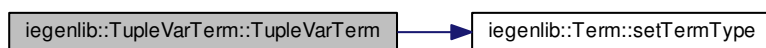
Here is the caller graph for this function:



### 9.15.2.2 iegenlib::TupleVarTerm::TupleVarTerm ( int *location* ) [inline]

Convenience constructor, assumes `coeff = 1`.

Here is the call graph for this function:



### 9.15.2.3 `iegenlib::TupleVarTerm::TupleVarTerm ( const TupleVarTerm & other )`

Copy constructor.

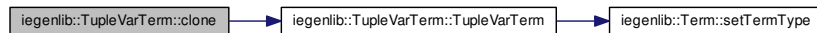
## 9.15.3 Member Function Documentation

### 9.15.3.1 `Term * iegenlib::TupleVarTerm::clone ( ) const` [virtual]

Create a copy of this [Term](#) (and of the same subclass)

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



### 9.15.3.2 `bool iegenlib::TupleVarTerm::factorMatches ( const Term & other ) const` [virtual]

Returns true if this term can be combined with the given term.

Returns true if this term has the same factor (i.e. everything except the coefficient) as the given other term.

Reimplemented from [iegenlib::Term](#).

Here is the caller graph for this function:



### 9.15.3.3 `bool iegenlib::TupleVarTerm::isConst( ) const [inline, virtual]`

Returns true if the [Term](#) is a const.

Reimplemented from [iegenlib::Term](#).

### 9.15.3.4 `bool iegenlib::TupleVarTerm::operator< ( const Term & other ) const [virtual]`

Comparison operator -- lexicographic order.

Compare two terms in a canonical order, defined as follows: 1. by term type: TupleVar, SymConst, UFCall, ConstVal 2. within each type, in alphabetical or numeric order.

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



### 9.15.3.5 `TupleVarTerm & iegenlib::TupleVarTerm::operator=( const TupleVarTerm & other )`

Copy assignment.

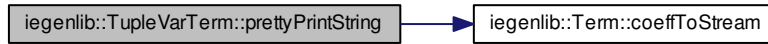
### 9.15.3.6 `std::string iegenlib::TupleVarTerm::prettyPrintString ( const std::vector< TupleElemDecl > & mTupleDecl, bool absValue = false ) const [virtual]`

Creates a compact string, pretty printed.

Creates a compact string, pretty printed (replace `__tv#` with name).

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



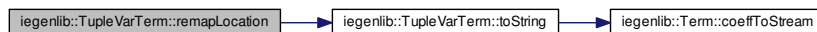
Here is the caller graph for this function:



9.15.3.7 `void iegenlib::TupleVarTerm::remapLocation ( const std::vector< int > & oldToNewLocs )`

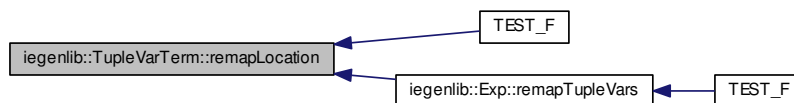
Remap our location according to the given map vector. See [Exp::remapTupleVars](#) for more detail.

Here is the call graph for this function:





Here is the caller graph for this function:



**9.15.3.8** `std::string iegenlib::TupleVarTerm::toString ( bool absValue = false ) const`  
`[virtual]`

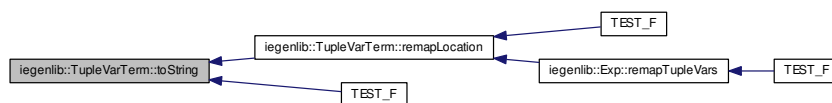
Creates a compact string to help with debugging.

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



Here is the caller graph for this function:



**9.15.3.9** `std::string iegenlib::TupleVarTerm::type ( ) const` `[virtual]`

Returns string of subclass type.

Reimplemented from [iegenlib::Term](#).

### 9.15.4 Member Data Documentation

9.15.4.1 int `iegenlib::TupleVarTerm::mLocation` [private]

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

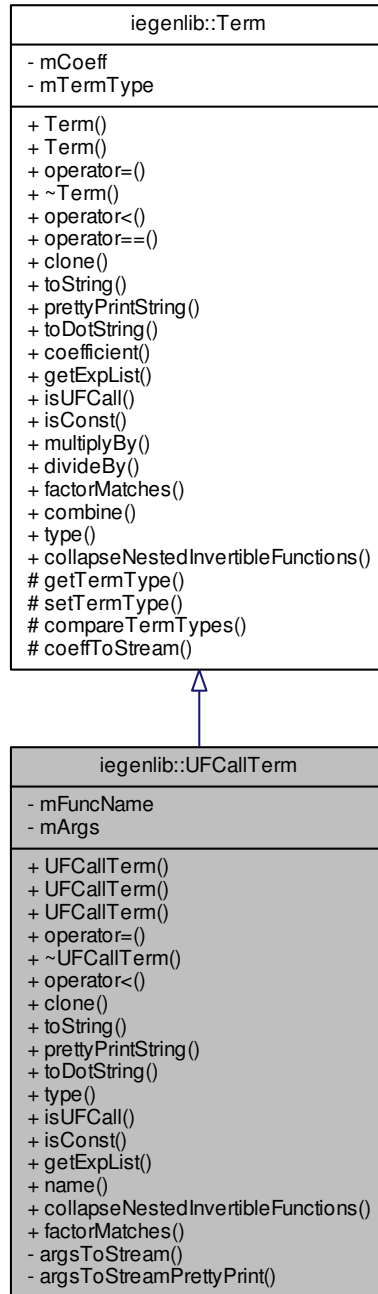
- [src/iegenlib/set\\_relation/expression.h](#)
- [src/iegenlib/set\\_relation/expression.cc](#)

## 9.16 `iegenlib::UFCallTerm` Class Reference

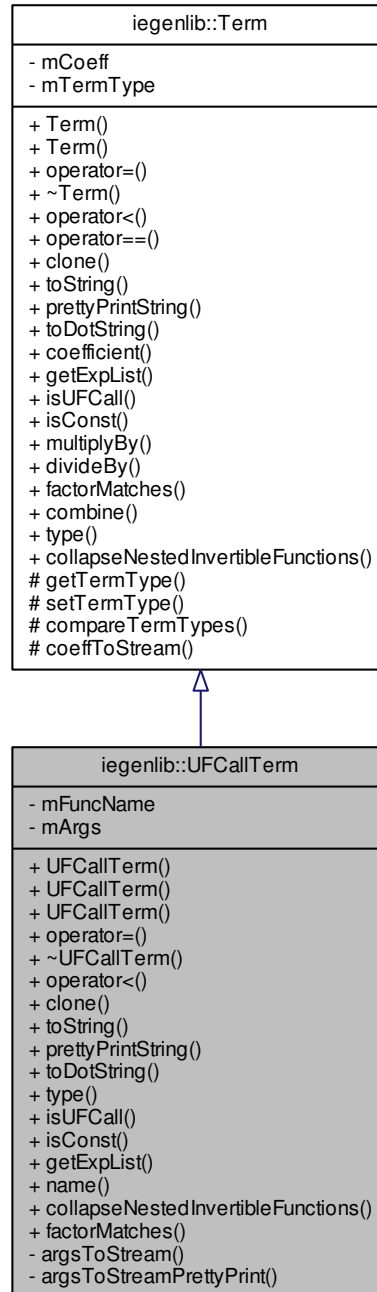
Represents a coefficient multiplied by an uninterpreted function call.

```
#include <expression.h>
```

Inheritance diagram for iegenlib::UFCallTerm:



Collaboration diagram for `iegenlib::UFCallTerm`:



## Public Member Functions

- [UFCallTerm](#) (int coeff, std::string funcName, std::list< [Exp](#) \* > \*args)
- [UFCallTerm](#) (std::string funcName, std::list< [Exp](#) \* > \*args)
  - Convenience constructor, assumes coeff = 1.*
- [UFCallTerm](#) (const [UFCallTerm](#) &other)
  - Copy constructor.*
- [UFCallTerm](#) & [operator=](#) (const [UFCallTerm](#) &other)
  - Copy assignment.*
- [~UFCallTerm](#) ()
  - Destructor.*
- bool [operator<](#) (const [Term](#) &other) const
  - Comparison operator -- lexicographic order.*
- [Term](#) \* [clone](#) () const
  - Create a copy of this [Term](#) (and of the same subclass)*
- std::string [toString](#) (bool absValue=false) const
  - Creates a compact string to help with debugging.*
- std::string [prettyPrintString](#) (const std::vector< [TupleElemDecl](#) > &mTupleDecl, bool absValue=false) const
  - Creates a compact string to help with debugging.*
- std::string [toDotString](#) (bool absValue=false) const
  - Creates a brief compact string to help with DOT output.*
- std::string [type](#) () const
  - Returns string of subclass type.*
- bool [isUFCall](#) () const
  - Returns true if the [Term](#) is really a [UFCallTerm](#).*
- bool [isConst](#) () const
  - Returns true if the [Term](#) is a const.*
- bool [getExpList](#) (std::list< [Exp](#) \* > \*\*outList)
- std::string [name](#) ()
  - Returns the function name as a string.*
- [Exp](#) \* [collapseNestedInvertibleFunctions](#) () const
- bool [factorMatches](#) (const [Term](#) &other) const
  - Returns true if this term can be combined with the given term.*

## Private Member Functions

- void [argsToStream](#) (std::stringstream &ss) const
- void [argsToStreamPrettyPrint](#) (const std::vector< [TupleElemDecl](#) > &mTupleDecl, std::stringstream &ss) const

## Private Attributes

- `std::string` `mFuncName`
- `std::list< Exp * >` `mArgs`

### 9.16.1 Detailed Description

Represents a coefficient multiplied by an uninterpreted function call.

### 9.16.2 Constructor & Destructor Documentation

#### 9.16.2.1 `iegenlib::UFCallTerm::UFCallTerm ( int coeff, std::string funcName, std::list< Exp * > * args )`

##### Constructor

Memory management: this object assumes ownership of the passed-in `Exp` objects. The caller must not destroy them, and should not assume they survive destruction of this `UFCallTerm`.

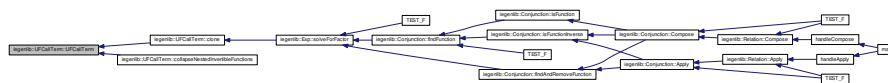
##### Parameters

<code>coeff</code>	-- coefficient for the call
<code>funcName</code>	-- function to call
<code>args</code>	-- arguments for the function call (adopted)

Here is the call graph for this function:



Here is the caller graph for this function:



### 9.16.2.2 iegenlib::UFCallTerm::UFCallTerm ( std::string *funcName*, std::list< Exp \* > \* *args* )

Convenience constructor, assumes coeff = 1.

Here is the call graph for this function:



### 9.16.2.3 iegenlib::UFCallTerm::UFCallTerm ( const UFCallTerm & *other* )

Copy constructor.

### 9.16.2.4 iegenlib::UFCallTerm::~~UFCallTerm ( )

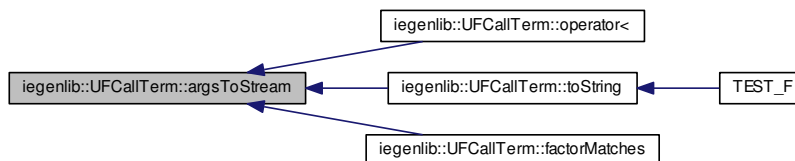
Destructor.

## 9.16.3 Member Function Documentation

### 9.16.3.1 void iegenlib::UFCallTerm::argsToStream ( std::stringstream & *ss* ) const [private]

Emits our argument list, as strings, to the given stream. Helper method for toString and operator<.

Here is the caller graph for this function:



9.16.3.2 `void iegenlib::UFCallTerm::argsToStreamPrettyPrint ( const std::vector< TupleElemDecl > & mTupleDecl, std::stringstream & ss ) const [private]`

Emits our argument list, as strings, to the given stream (pretty printed). Helper method for prettyPrintString.

Here is the caller graph for this function:

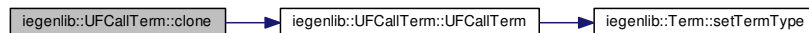


9.16.3.3 `Term * iegenlib::UFCallTerm::clone ( ) const [virtual]`

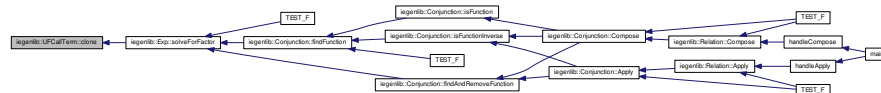
Create a copy of this [Term](#) (and of the same subclass)

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



Here is the caller graph for this function:



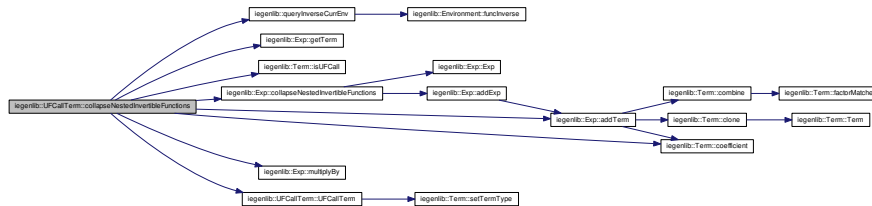
9.16.3.4 `Exp * iegenlib::UFCallTerm::collapseNestedInvertibleFunctions ( ) const [virtual]`

Return a new [Exp](#) with all nested functions such as `f ( f_inv ( i ) )` changed to `i`.



Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:

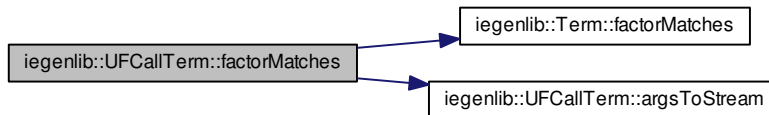


#### 9.16.3.5 `bool iegenlib::UFCallTerm::factorMatches ( const Term & other ) const` [virtual]

Returns true if this term can be combined with the given term.

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:

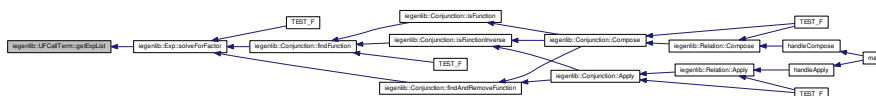


#### 9.16.3.6 `bool iegenlib::UFCallTerm::getExpList ( std::list< Exp * > ** outList )` [virtual]

Get a list of any expressions contained by this term. this function does not make a copy, so if you delete the containing object and then try to use any thing from the list you will get a memory error

Reimplemented from [iegenlib::Term](#).

Here is the caller graph for this function:



### 9.16.3.7 bool iegenlib::UFCallTerm::isConst ( ) const [inline, virtual]

Returns true if the `Term` is a const.

Reimplemented from `iegenlib::Term`.

### 9.16.3.8 bool iegenlib::UFCallTerm::isUFCall ( ) const [inline, virtual]

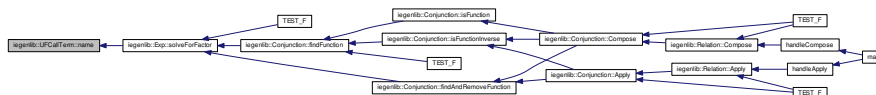
Returns true if the `Term` is really a `UFCallTerm`.

Reimplemented from `iegenlib::Term`.

### 9.16.3.9 std::string iegenlib::UFCallTerm::name ( ) [inline]

Returns the function name as a string.

Here is the caller graph for this function:



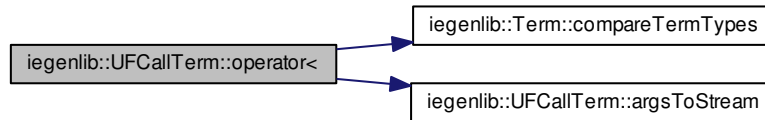
### 9.16.3.10 bool iegenlib::UFCallTerm::operator< ( const Term & other ) const [virtual]

Comparison operator -- lexicographic order.

Compare two terms in a canonical order, defined as follows: 1. by term type: `TupleVar`, `SymConst`, `UFCall`, `ConstVal` 2. within each type, in alphabetical or numeric order.

Reimplemented from `iegenlib::Term`.

Here is the call graph for this function:



#### 9.16.3.11 `UFCallTerm & iegenlib::UFCallTerm::operator= ( const UFCallTerm & other )`

Copy assignment.

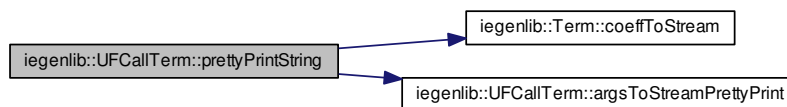
#### 9.16.3.12 `std::string iegenlib::UFCallTerm::prettyPrintString ( const std::vector< TupleElemDecl > & mTupleDecl, bool absValue = false ) const [virtual]`

Creates a compact string to help with debugging.

Creates a compact string, pretty Printed.

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



#### 9.16.3.13 `std::string iegenlib::UFCallTerm::toDotString ( bool absValue = false ) const [virtual]`

Creates a brief compact string to help with DOT output.

Creates a compact string to help with DOT output.

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:

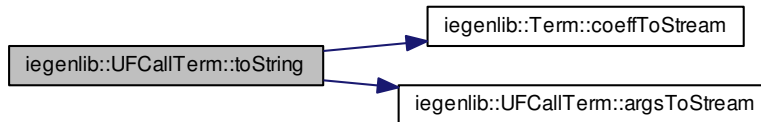


9.16.3.14 `std::string iegenlib::UFCallTerm::toString ( bool absValue = false ) const`  
`[virtual]`

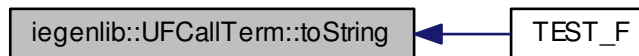
Creates a compact string to help with debugging.

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



Here is the caller graph for this function:



9.16.3.15 `std::string iegenlib::UFCallTerm::type ( ) const` [virtual]

Returns string of subclass type.

Reimplemented from [iegenlib::Term](#).

#### 9.16.4 Member Data Documentation

9.16.4.1 `std::list<Exp*> iegenlib::UFCallTerm::mArgs` [private]

9.16.4.2 `std::string iegenlib::UFCallTerm::mFuncName` [private]

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

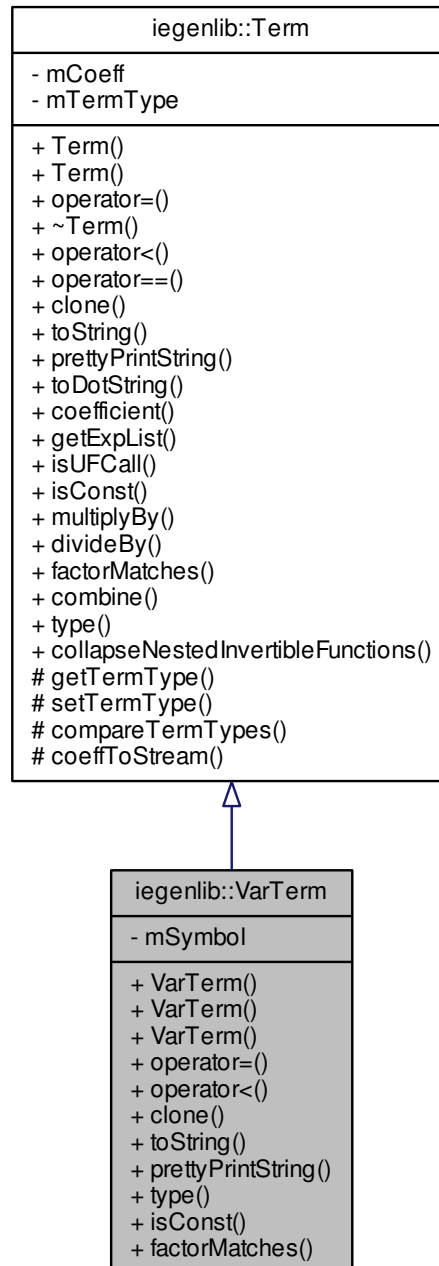
- [src/iegenlib/set\\_relation/expression.h](#)
- [src/iegenlib/set\\_relation/expression.cc](#)

## 9.17 `iegenlib::VarTerm` Class Reference

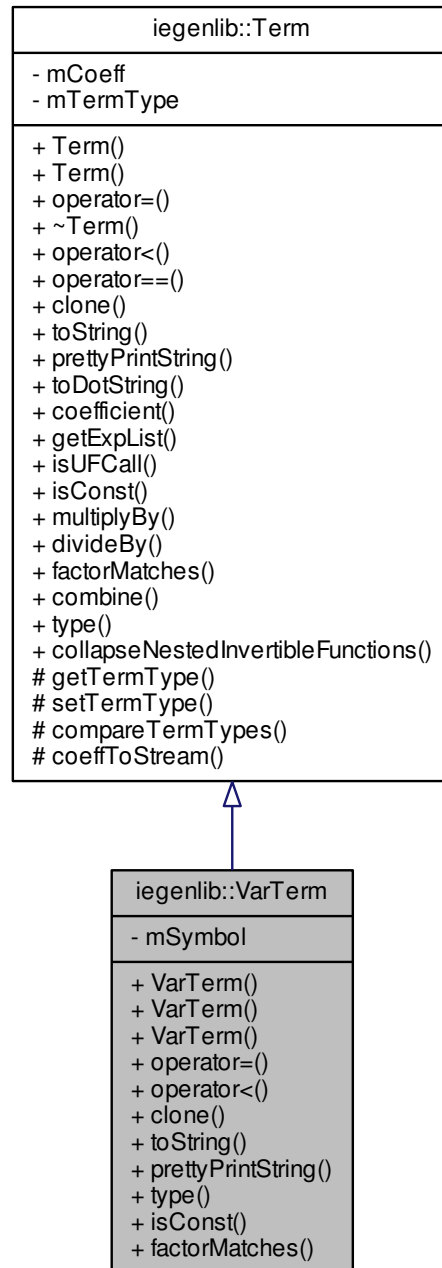
Represents a coefficient multiplied by a variable or symbolic constant.

```
#include <expression.h>
```

Inheritance diagram for `iegenlib::VarTerm`:



Collaboration diagram for iegenlib::VarTerm:



## Public Member Functions

- [VarTerm](#) (int coeff, std::string symbol)  
*Constructor.*
- [VarTerm](#) (std::string symbol)  
*Convenience constructor, assumes coeff = 1.*
- [VarTerm](#) (const [VarTerm](#) &other)  
*Copy constructor.*
- [VarTerm](#) & [operator=](#) (const [VarTerm](#) &other)  
*Copy assignment.*
- bool [operator<](#) (const [Term](#) &other) const  
*Comparison operator -- lexicographic order.*
- [Term](#) \* [clone](#) () const  
*Create a copy of this [Term](#) (and of the same subclass)*
- std::string [toString](#) (bool absValue=false) const  
*Creates a compact string to help with debugging.*
- std::string [prettyPrintString](#) (const std::vector< [TupleElemDecl](#) > &mTupleDecl, bool absValue=false) const  
*Creates a compact string, pretty printed.*
- std::string [type](#) () const  
*Returns string of subclass type.*
- bool [isConst](#) () const  
*Returns true if the [Term](#) is a const.*
- bool [factorMatches](#) (const [Term](#) &other) const  
*Returns true if this term can be combined with the given term.*

## Private Attributes

- std::string [mSymbol](#)

### 9.17.1 Detailed Description

Represents a coefficient multiplied by a variable or symbolic constant.

### 9.17.2 Constructor & Destructor Documentation

9.17.2.1 `iegenlib::VarTerm::VarTerm ( int coeff, std::string symbol )` [`inline`]

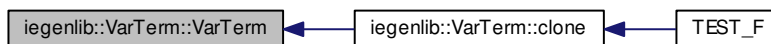
Constructor.



Here is the call graph for this function:



Here is the caller graph for this function:



#### 9.17.2.2 `iegenlib::VarTerm::VarTerm ( std::string symbol ) [inline]`

Convenience constructor, assumes `coeff = 1`.

Here is the call graph for this function:



#### 9.17.2.3 `iegenlib::VarTerm::VarTerm ( const VarTerm & other )`

Copy constructor.

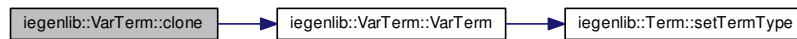
### 9.17.3 Member Function Documentation

#### 9.17.3.1 `Term * iegenlib::VarTerm::clone ( ) const` [virtual]

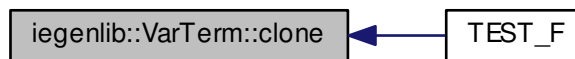
Create a copy of this [Term](#) (and of the same subclass)

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



Here is the caller graph for this function:

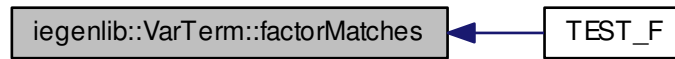


#### 9.17.3.2 `bool iegenlib::VarTerm::factorMatches ( const Term & other ) const` [virtual]

Returns true if this term can be combined with the given term.

Reimplemented from [iegenlib::Term](#).

Here is the caller graph for this function:



#### 9.17.3.3 bool iegenlib::VarTerm::isConst ( ) const [inline, virtual]

Returns true if the [Term](#) is a const.

Reimplemented from [iegenlib::Term](#).

#### 9.17.3.4 bool iegenlib::VarTerm::operator< ( const Term & other ) const [virtual]

Comparison operator -- lexicographic order.

Compare two terms in a canonical order, defined as follows: 1. by term type: TupleVar, SymConst, UFCall, ConstVal 2. within each type, in alphabetical or numeric order.

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



#### 9.17.3.5 VarTerm & iegenlib::VarTerm::operator= ( const VarTerm & other )

Copy assignment.

9.17.3.6 `std::string iegenlib::VarTerm::prettyPrintString ( const std::vector< TupleElemDecl > & mTupleDecl, bool absValue = false ) const` [virtual]

Creates a compact string, pretty printed.

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



9.17.3.7 `std::string iegenlib::VarTerm::toString ( bool absValue = false ) const` [virtual]

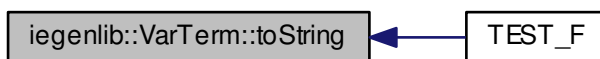
Creates a compact string to help with debugging.

Reimplemented from [iegenlib::Term](#).

Here is the call graph for this function:



Here is the caller graph for this function:



9.17.3.8 `std::string iegenlib::VarTerm::type( ) const` [virtual]

Returns string of subclass type.

Reimplemented from [iegenlib::Term](#).

## 9.17.4 Member Data Documentation

9.17.4.1 `std::string iegenlib::VarTerm::mSymbol` [private]

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

- [src/iegenlib/set\\_relation/expression.h](#)
- [src/iegenlib/set\\_relation/expression.cc](#)

## 9.18 yy\_buffer\_state Struct Reference

### Public Attributes

- FILE \* [yy\\_input\\_file](#)
- char \* [yy\\_ch\\_buf](#)
- char \* [yy\\_buf\\_pos](#)
- [yy\\_size\\_t](#) [yy\\_buf\\_size](#)
- int [yy\\_n\\_chars](#)
- int [yy\\_is\\_our\\_buffer](#)
- int [yy\\_is\\_interactive](#)
- int [yy\\_at\\_bol](#)
- int [yy\\_bs\\_lineno](#)

- int [yy\\_bs\\_column](#)
- int [yy\\_fill\\_buffer](#)
- int [yy\\_buffer\\_status](#)

### 9.18.1 Member Data Documentation

9.18.1.1 int `yy_buffer_state::yy_at_bol`

9.18.1.2 int `yy_buffer_state::yy_bs_column`

The column count.

9.18.1.3 int `yy_buffer_state::yy_bs_lineno`

The line count.

9.18.1.4 char\* `yy_buffer_state::yy_buf_pos`

9.18.1.5 `yy_size_t` `yy_buffer_state::yy_buf_size`

9.18.1.6 int `yy_buffer_state::yy_buffer_status`

9.18.1.7 char\* `yy_buffer_state::yy_ch_buf`

9.18.1.8 int `yy_buffer_state::yy_fill_buffer`

9.18.1.9 FILE\* `yy_buffer_state::yy_input_file`

9.18.1.10 int `yy_buffer_state::yy_is_interactive`

9.18.1.11 int `yy_buffer_state::yy_is_our_buffer`

9.18.1.12 int `yy_buffer_state::yy_n_chars`

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

- `src/iegenlib/parser/gen_scanner.cc`

## 9.19 `yy_trans_info` Struct Reference

## Public Attributes

- [flex\\_int32\\_t yy\\_verify](#)
- [flex\\_int32\\_t yy\\_nxt](#)

### 9.19.1 Member Data Documentation

9.19.1.1 [flex\\_int32\\_t yy\\_trans\\_info::yy\\_nxt](#)

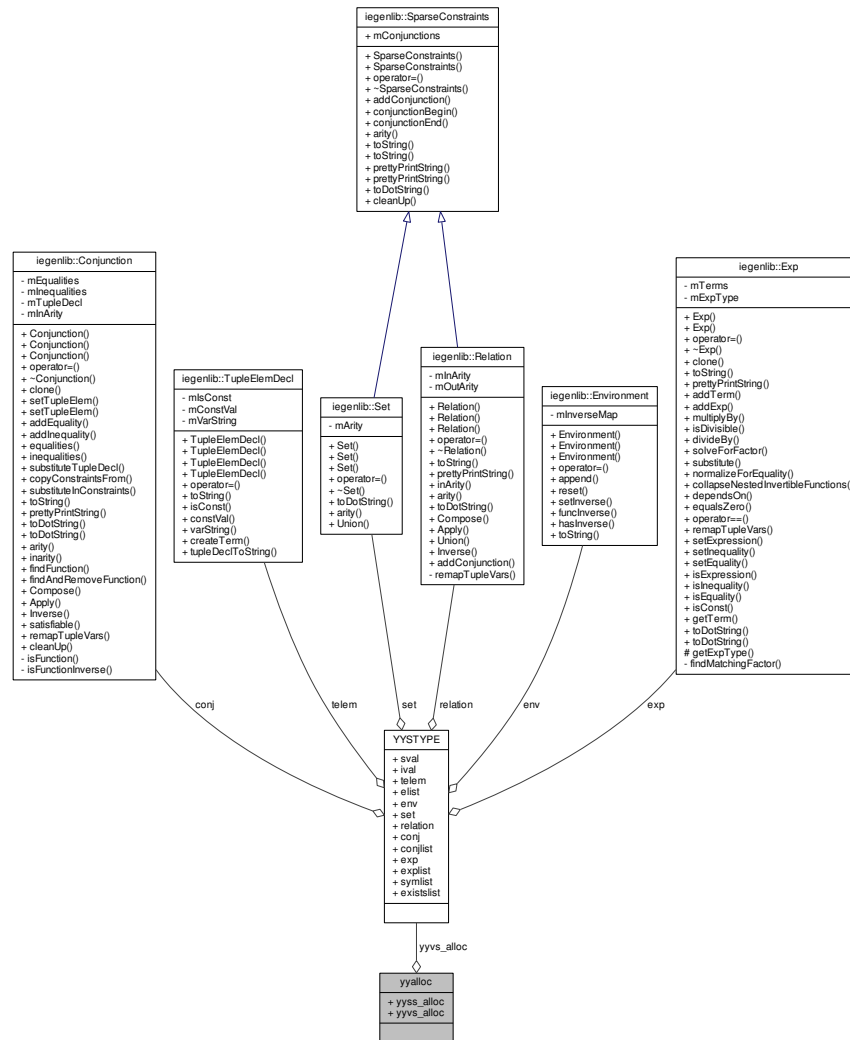
9.19.1.2 [flex\\_int32\\_t yy\\_trans\\_info::yy\\_verify](#)

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

- [src/iegenlib/parser/gen\\_scanner.cc](#)

## 9.20 yyalloc Union Reference

Collaboration diagram for yyalloc:



### Public Attributes

- `yytype_int16 yyss_alloc`



- [YYSTYPE yyvs\\_alloc](#)

### 9.20.1 Member Data Documentation

#### 9.20.1.1 yytype\_int16 yyalloc::yyss\_alloc

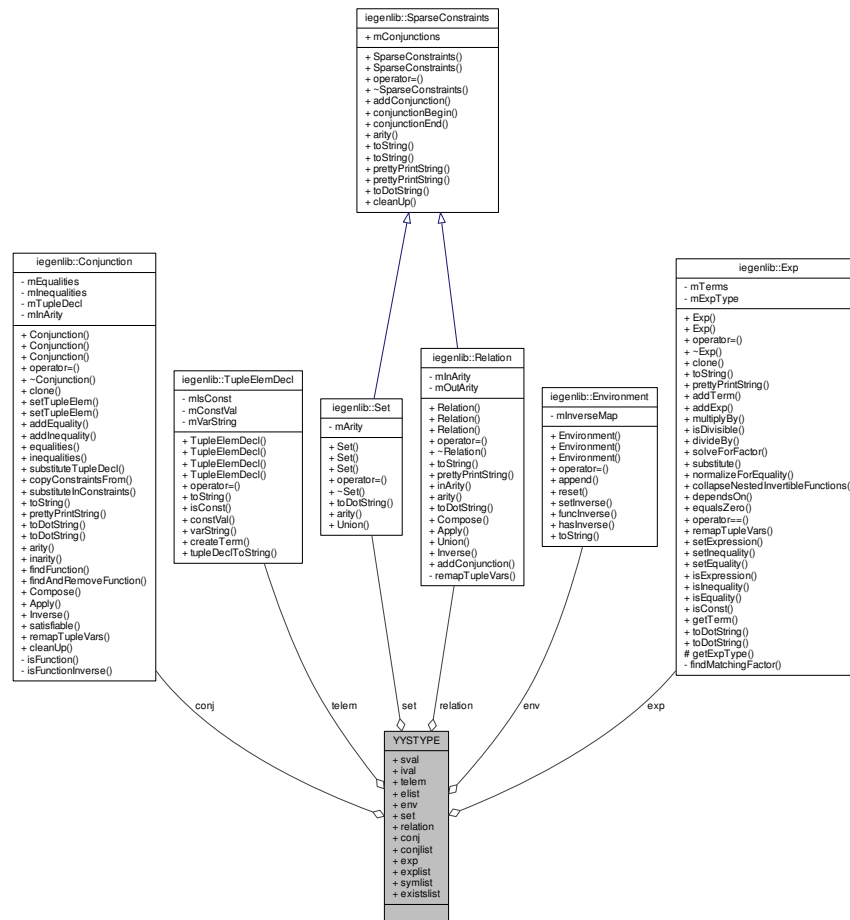
#### 9.20.1.2 YYSTYPE yyalloc::yyvs\_alloc

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

- [src/iegenlib/parser/gen\\_parser.cc](#)

## 9.21 YYSTYPE Union Reference

Collaboration diagram for YYSTYPE:



### Public Attributes

- `std::string * sval`
- `int ival`
- `iegenlib::TupleElemDecl * telem`
- `std::list < iegenlib::TupleElemDecl * > * elist`
- `iegenlib::Environment * env`

- [iegenlib::Set](#) \* [set](#)
- [iegenlib::Relation](#) \* [relation](#)
- [iegenlib::Conjunction](#) \* [conj](#)
- `std::list` < [iegenlib::Conjunction](#) \* > \* [conjlist](#)
- [iegenlib::Exp](#) \* [exp](#)
- `std::list` < [iegenlib::Exp](#) \* > \* [explist](#)
- `std::list` < `std::string` > \* [symlist](#)
- `std::list` < `std::string` > \* [existslist](#)

### 9.21.1 Member Data Documentation

9.21.1.1 [iegenlib::Conjunction](#)\* YYSTYPE::conj

9.21.1.2 `std::list`<[iegenlib::Conjunction](#)\*>\* YYSTYPE::conjlist

9.21.1.3 `std::list`<[iegenlib::TupleElemDecl](#)\*>\* YYSTYPE::elist

9.21.1.4 [iegenlib::Environment](#)\* YYSTYPE::env

9.21.1.5 `std::list`<`std::string`>\* YYSTYPE::existslist

9.21.1.6 [iegenlib::Exp](#)\* YYSTYPE::exp

9.21.1.7 `std::list`<[iegenlib::Exp](#)\*>\* YYSTYPE::explist

9.21.1.8 `int` YYSTYPE::ival

9.21.1.9 [iegenlib::Relation](#)\* YYSTYPE::relation

9.21.1.10 [iegenlib::Set](#)\* YYSTYPE::set

9.21.1.11 `std::string`\* YYSTYPE::sval

9.21.1.12 `std::list`<`std::string`>\* YYSTYPE::symlist

9.21.1.13 [iegenlib::TupleElemDecl](#)\* YYSTYPE::telem

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

- [src/iegenlib/parser/gen\\_parser.cc](#)



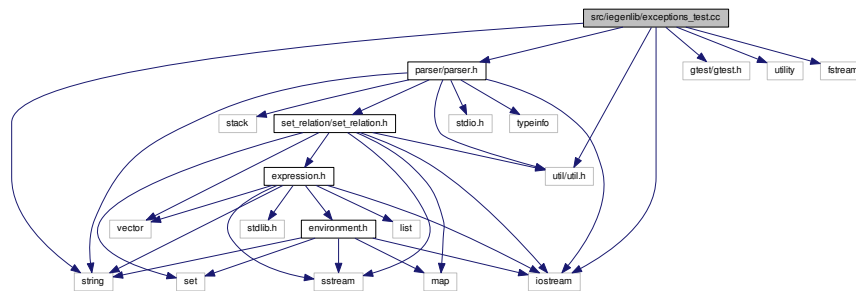
# Chapter 10

## File Documentation

### 10.1 src/iegenlib/exceptions\_test.cc File Reference

Test exceptions for the parser/, set\_relation/ and util/ classes.

```
#include <parser/parser.h> #include <util/util.h> #include  
<gtest/gtest.h> #include <utility> #include <fstream> ×  
#include <iostream> #include <string> Include dependency graph  
for exceptions_test.cc:
```



### Classes

- class [ExceptionTestParser](#)  
*Class to test exception handling in parser/ methods.*
- class [ExceptionTestSetRelation](#)  
*Class to test exception handling in set\_relation/ methods.*

- class [ExceptionTestExp](#)

*Class to test exception handling in expression/ methods.*

## Functions

- [TEST\\_F](#) ([ExceptionTestParser](#), [ParseSetMethod](#))
- [TEST\\_F](#) ([ExceptionTestParser](#), [ParseRelationMethod](#))
- [TEST\\_F](#) ([ExceptionTestParser](#), [ParseEnvMethod](#))
- [TEST\\_F](#) ([ExceptionTestSetRelation](#), [SetTupleElemMethod](#))
- [TEST\\_F](#) ([ExceptionTestSetRelation](#), [ComposeMethod](#))
- [TEST\\_F](#) ([ExceptionTestSetRelation](#), [ApplyMethod](#))
- [TEST\\_F](#) ([ExceptionTestSetRelation](#), [UnionMethod](#))
- [TEST\\_F](#) ([ExceptionTestExp](#), [PrettyPrintStringMethod](#))
- [TEST\\_F](#) ([ExceptionTestExp](#), [RemapLocationMethod](#))
- [TEST\\_F](#) ([ExceptionTestExp](#), [FindMatchingFactorMethod](#))
- [TEST\\_F](#) ([ExceptionTestExp](#), [SolveForFactorMethod](#))

### 10.1.1 Detailed Description

Test exceptions for the `parser/`, `set_relation/` and `util/` classes. This file is to test all of the `Exception`, along with the related classes: `EqExp`, `IneqExp`, `UFCallTerm`, `VarTerm`, and `TupleVarTerm`.

#### Date

Started: 5/30/12 #

#### Revision:

391

: last committed revision #

#### Date:

5/30/12

: date of last committed revision #

#### Author:

heim

: author of last committed revision

Authors

Mark Heim

Copyright (c) 2012, Colorado State University

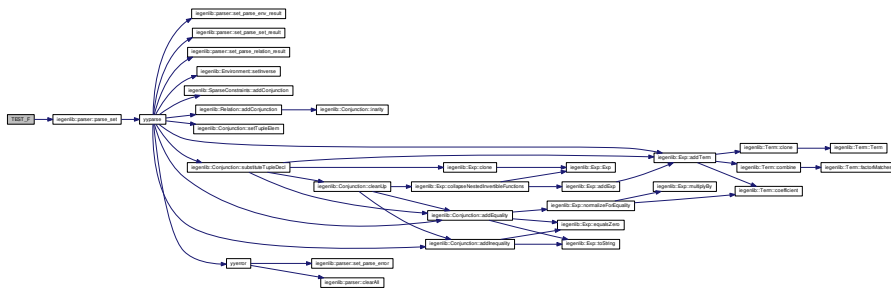
All rights reserved.

See ../COPYING for details.

10.1.2 Function Documentation

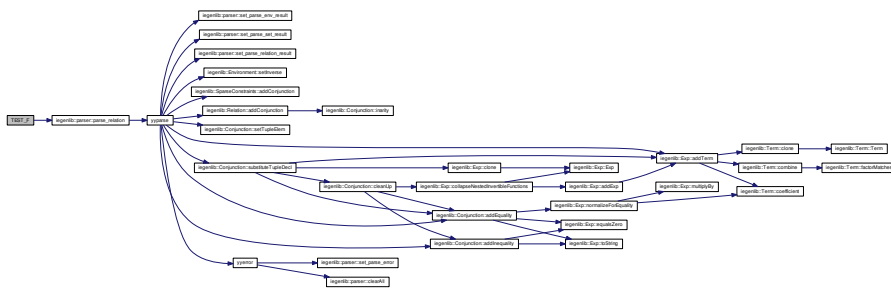
10.1.2.1 TEST\_F ( ExceptionTestParser , ParseSetMethod )

Here is the call graph for this function:



10.1.2.2 TEST\_F ( ExceptionTestParser , ParseRelationMethod )

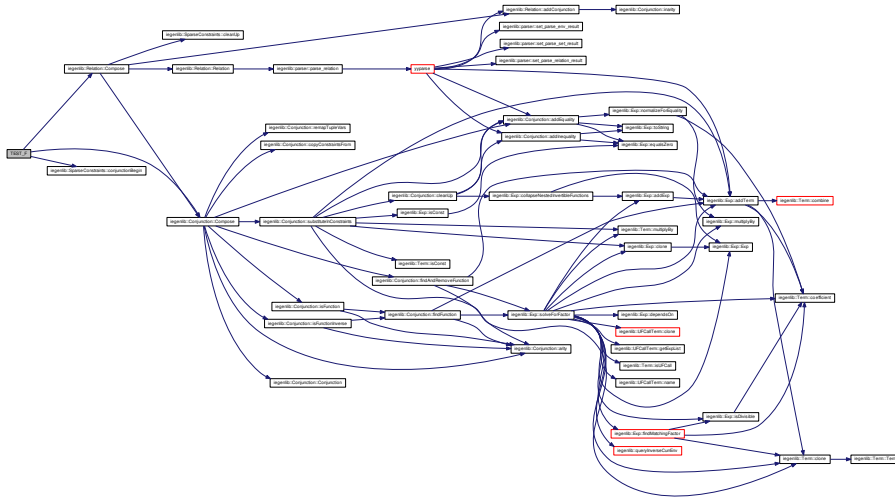
Here is the call graph for this function:







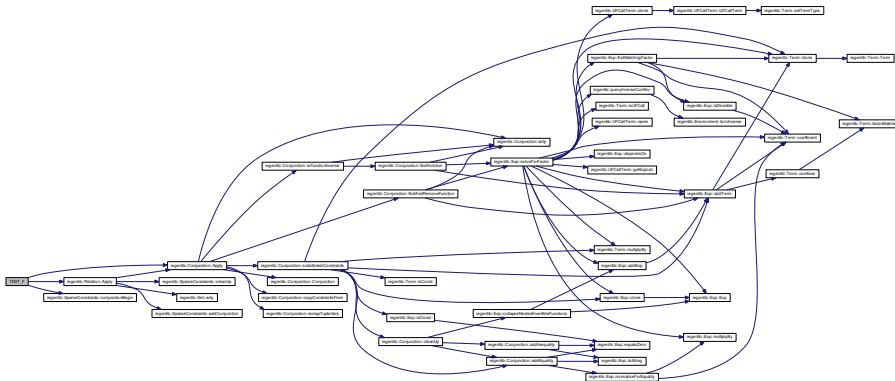
Here is the call graph for this function:



10.1.2.6 TEST\_F ( ExceptionTestSetRelation , ApplyMethod )

Tests exception handling in method with following signatures in [set\\_relation.cc](#): Set \*Relation::Apply(const Set\* rhs) const Conjunction \*Conjunction::Apply(const Conjunction \*rhs) const These methods contain two throw statements.

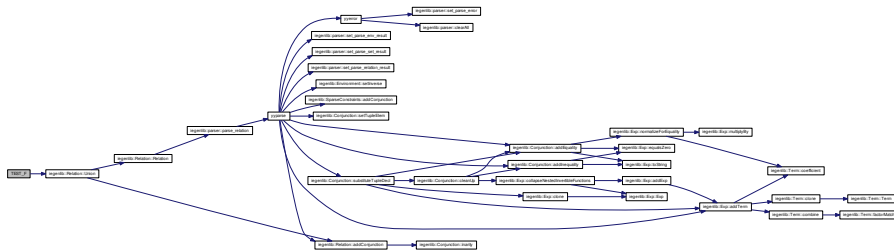
Here is the call graph for this function:



### 10.1.2.7 TEST\_F ( ExceptionTestSetRelation , UnionMethod )

Tests exception handling in method with following signatures in [set\\_relation.cc](#):  
 - Relation\* Relation::Union(const Relation\* rhs) const Set\* Set::Union(const Set\* rhs) const  
 These methods contain two throw statements.

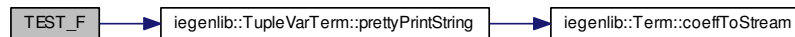
Here is the call graph for this function:



### 10.1.2.8 TEST\_F ( ExceptionTestExp , PrettyPrintStringMethod )

Tests exception handling in method with following signature in [expression.cc](#):  
 std::string TupleVarTerm::prettyPrintString( const std::vector<TupleElemDecl>& mTupleDecl, bool absValue) const  
 This method contains one throw statement.

Here is the call graph for this function:



### 10.1.2.9 TEST\_F ( ExceptionTestExp , RemapLocationMethod )

Tests exception handling in method with following signature in [expression.cc](#):  
 void TupleVarTerm::remapLocation(const std::vector<int>& oldToNewLocs)  
 This method contains two throw statements.

Here is the call graph for this function:



#### 10.1.2.10 TEST\_F ( ExceptionTestExp , FindMatchingFactorMethod )

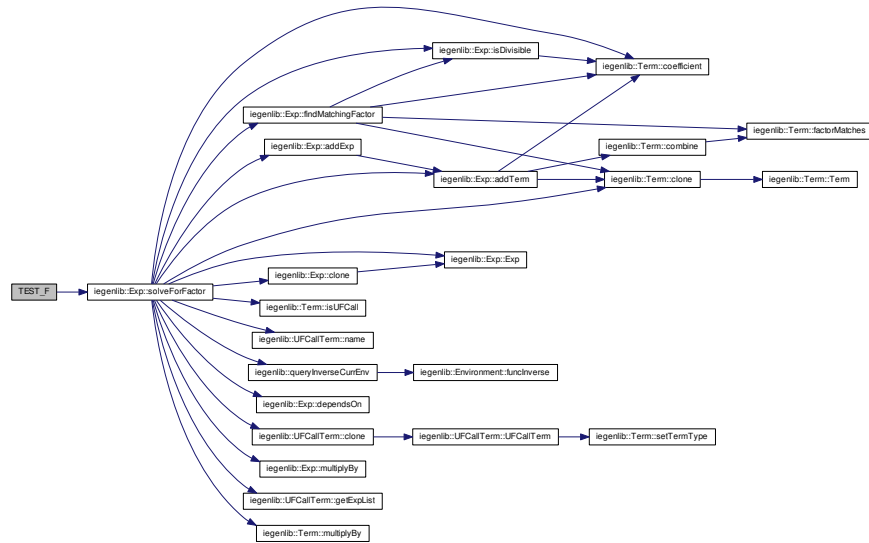
Tests exception handling in method with following signature in [expression.cc](#): Term\* Exp::findMatchingFactor(const Term & factor) const This method contains one throw statement.

This exception is unreachable in the code unless findMatchingFactor is changed from being a private method into a public or protected method.

#### 10.1.2.11 TEST\_F ( ExceptionTestExp , SolveForFactorMethod )

Tests exception handling in method with following signature in [expression.cc](#): Exp\* - Exp::solveForFactor(Term\* factor, const Environment \*env) const This method contains one throw statement.

Here is the call graph for this function:



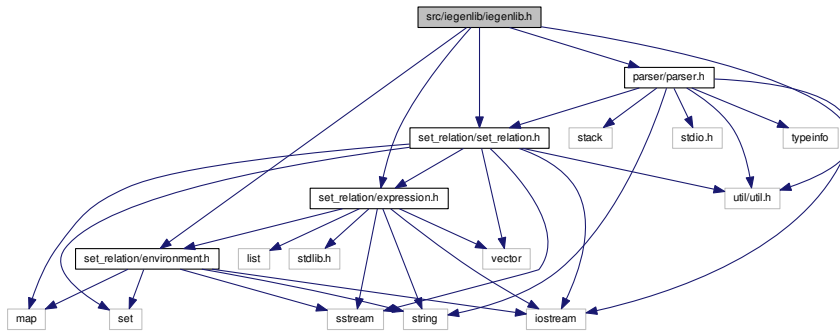
## 10.2 src/iegenlib/iegenlib.h File Reference

Main IEGen header file.

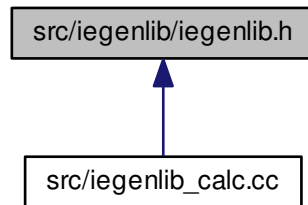
```

#include <set_relation/environment.h>      #include <set_
relation/expression.h> #include <set_relation/set_relation.
h> #include <util/util.h> #include <parser/parser.h> Include
  
```

dependency graph for iegenlib.h:



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



### 10.2.1 Detailed Description

Main IEGen header file.

#### Date

Started: 11/15/2010 #

#### Revision:

457

: last committed revision #

Date:

2012-06-28 13:41:00 -0#

: date of last committed revision #

Author:

mstrout

: author of last committed revision

Authors

Alan LaMielle

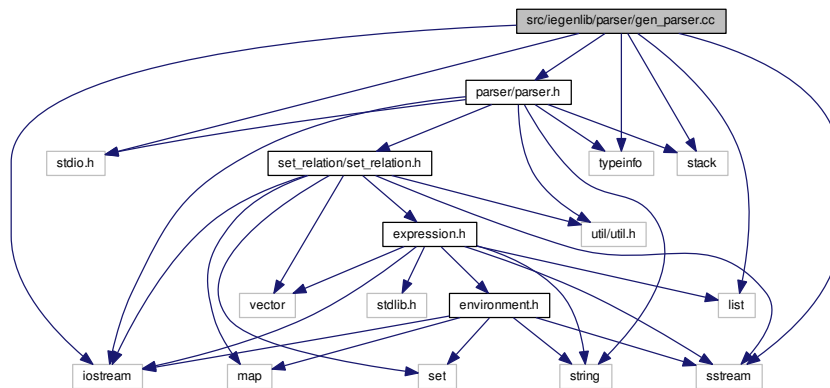
Copyright (c) 2009, 2010, 2011, 2012, Colorado State University

All rights reserved.

See ../COPYING for details.

### 10.3 src/iegenlib/parser/gen\_parser.cc File Reference

```
#include <parser/parser.h> #include <iostream> #include
<list> #include <stack> #include <stdio.h> #include <typeinfo> ×
#include <sstream> Include dependency graph for gen_parser.cc:
```



## Classes

- union [YYSTYPE](#)
- union [yyallocc](#)

## Defines

- #define [YYBISON](#) 1
- #define [YYBISON\\_VERSION](#) "2.5"
- #define [YYSKELETON\\_NAME](#) "yacc.c"
- #define [YYPURE](#) 0
- #define [YYPUSH](#) 0
- #define [YYPULL](#) 1
- #define [YYLSP\\_NEEDED](#) 0
- #define [YYERROR\\_VERBOSE](#)
- #define [YYDEBUG](#) 1
- #define [YYERROR\\_VERBOSE](#) 1
- #define [YYTOKEN\\_TABLE](#) 0
- #define [YYTOKENTYPE](#)
- #define [YYSTYPE\\_IS\\_TRIVIAL](#) 1
- #define [yystate](#) YYSTYPE
- #define [YYSTYPE\\_IS\\_DECLARED](#) 1
- #define [YYSIZE\\_T](#) unsigned int
- #define [YYSIZE\\_MAXIMUM](#) ((YYSIZE\_T) -1)
- #define [YY\\_\(msgid\)](#) msgid
- #define [YYUSE\(e\)](#) ((void) (e))
- #define [YYID\(n\)](#) (n)
- #define [YYSTACK\\_ALLOC](#) YYMALLOC
- #define [YYSTACK\\_FREE](#) YYFREE
- #define [YYSTACK\\_ALLOC\\_MAXIMUM](#) YYSIZE\_MAXIMUM
- #define [YYMALLOC](#) malloc
- #define [YYFREE](#) free
- #define [YYSTACK\\_GAP\\_MAXIMUM](#) (sizeof (union [yyallocc](#)) - 1)
- #define [YYSTACK\\_BYTES\(N\)](#)
- #define [YYCOPY\\_NEEDED](#) 1
- #define [YYSTACK\\_RELOCATE](#)(Stack\_alloc, Stack)
- #define [YYCOPY](#)(To, From, Count)
- #define [YYFINAL](#) 14
- #define [YYLAST](#) 115
- #define [YYNTOKENS](#) 31
- #define [YYNNTS](#) 39
- #define [YYNRULES](#) 74

- #define `YYNSTATES` 137
- #define `YYUNDEFTOK` 2
- #define `YYMAXUTOK` 285
- #define `YYTRANSLATE(YYX)` `((unsigned int) (YYX) <= YYMAXUTOK ? yytranslate[YYX] : YYUNDEFTOK)`
- #define `YYPACT_NINF` -58
- #define `YYTABLE_NINF` -39
- #define `yypact_value_is_default(yystate)` `((yystate) == (-58))`
- #define `yytable_value_is_error(yytable_value)` `YYID (0)`
- #define `yyerrok` `(yyerrstatus = 0)`
- #define `yyclearin` `(yychar = YYEMPTY)`
- #define `YYEMPTY` (-2)
- #define `YYEOF` 0
- #define `YYACCEPT` `goto yyacceptlab`
- #define `YYABORT` `goto yyabortlab`
- #define `YYERROR` `goto yyerrorlab`
- #define `YYFAIL` `goto yyerrlab`
- #define `YYRECOVERING()` `(!!yyerrstatus)`
- #define `YYBACKUP` `(Token, Value)`
- #define `YYTERROR` 1
- #define `YYERRCODE` 256
- #define `YYRHSLOC` `(Rhs, K)` `((Rhs)[K])`
- #define `YYLOC_DEFAULT` `(Current, Rhs, N)`
- #define `YY_LOCATION_PRINT` `(File, Loc)` `((void) 0)`
- #define `YYLEX` `yylex ()`
- #define `YYFPRINTF` `fprintf`
- #define `YYDPRINTF` `(Args)`
- #define `YY_SYMBOL_PRINT` `(Title, Type, Value, Location)`
- #define `YY_STACK_PRINT` `(Bottom, Top)`
- #define `YY_REDUCE_PRINT` `(Rule)`
- #define `YYINITDEPTH` 200
- #define `YYMAXDEPTH` 10000
- #define `YYCASE_(N, S)`
- #define `YYPOPSTACK(N)` `(yyvsp -= (N), yyssp -= (N))`
- #define `YYSYNTAX_ERROR`

## Typedefs

- typedef union `YYSTYPE` `YYSTYPE`
- typedef unsigned char `yytype_uint8`
- typedef short int `yytype_int8`
- typedef unsigned short int `yytype_uint16`
- typedef short int `yytype_int16`



## Enumerations

- enum `yytokentype` { `LBRACE` = 258, `RBRACE` = 259, `LBRACKET` = 260, `RBRACKET` = 261, `LPAREN` = 262, `RPAREN` = 263, `COMMA` = 264, `COLON` = 265, `LT` = 266, `LTE` = 267, `GT` = 268, `GTE` = 269, `SEMI` = 270, `OR` = 271, `UNION` = 272, `INVERSE` = 273, `EXISTS` = 274, `EQ` = 275, `ARROW` = 276, `ID` = 277, `INT` = 278, `INVALID_ID` = 279, `DASH` = 280, `PLUS` = 281, `STAR` = 282, `UMINUS` = 283, `WAND` = 284, `AND` = 285 }

## Functions

- int `yylex` ()
- void `yyerror` (const char \*)
- static void `yy_symbol_value_print` (yyoutput, `yytype`, `yyvaluep`) FILE \*yyoutput
- static YYSIZE\_T `yystrlen` (yystr) const char \*yystr
- int `yyvsparse` ()

## Variables

- char \* `yytext`
- int `yylineno` = 1
- static const `yytype_uint8` `yytranslate` []
- static const `yytype_uint8` `yyprhs` []
- static const `yytype_int8` `yyrhs` []
- static const `yytype_uint16` `yyrline` []
- static const char \*const `yytname` []
- static const `yytype_uint8` `yyr1` []
- static const `yytype_uint8` `yyr2` []
- static const `yytype_uint8` `yydefact` []
- static const `yytype_int8` `yydefgoto` []
- static const `yytype_int8` `yyfact` []
- static const `yytype_int8` `yygoto` []
- static const `yytype_int16` `yytable` []
- static const `yytype_uint8` `yycheck` []
- static const `yytype_uint8` `yystos` []
- int `yytype`
- `YYSTYPE` const \*const `yyvaluep`
- `yytype_int16` \* `yytop`
- int `yyrule`
- const char \* `yysrc`
- int `yychar`
- `YYSTYPE` `yylval`
- int `yynerres`

### 10.3.1 Define Documentation

10.3.1.1 **#define YY\_( *msgid* )** *msgid*

10.3.1.2 **#define YY\_LOCATION\_PRINT( *File*, *Loc* )**((void) 0)

10.3.1.3 **#define YY\_REDUCE\_PRINT( *Rule* )**

**Value:**

```
do {
    if (yydebug)
        yy_reduce_print (yyvsp, Rule);
} while (YYID (0))
```

10.3.1.4 **#define YY\_STACK\_PRINT( *Bottom*, *Top* )**

**Value:**

```
do {
    if (yydebug)
        yy_stack_print ((Bottom), (Top));
} while (YYID (0))
```

10.3.1.5 **#define YY\_SYMBOL\_PRINT( *Title*, *Type*, *Value*, *Location* )**

**Value:**

```
do {
    if (yydebug)
    {
        YYFPRINTF (stderr, "%s ", Title);
        yy_symbol_print (stderr,
            Type, Value);
        YYFPRINTF (stderr, "\n");
    }
} while (YYID (0))
```

10.3.1.6 **#define YYABORT** goto yyabortlab

10.3.1.7 **#define YYACCEPT** goto yyacceptlab

10.3.1.8 **#define YYBACKUP( *Token*, *Value* )**

**Value:**

```

do
    if (yychar == YYEMPTY && yylen == 1)
    {
        yychar = (Token);
        yyval = (Value);
        YYPOPSTACK (1);
        goto yybackup;
    }
    else
    {
        yyerror (YY_("syntax error: cannot back up")); \
        YYERROR; \
    }
while (YYID (0))

```

**10.3.1.9 #define YBISON 1**

**10.3.1.10 #define YBISON\_VERSION "2.5"**

**10.3.1.11 #define YYCASE\_( N, S )**

**Value:**

```

case N:
    yyformat = S;
    break

```

**10.3.1.12 #define yyclearin (yychar = YYEMPTY)**

**10.3.1.13 #define YYCOPY( To, From, Count )**

**Value:**

```

do
    {
        YYSIZE_T yyi;
        for (yyi = 0; yyi < (Count); yyi++)
            (To)[yyi] = (From)[yyi];
    }
while (YYID (0))

```

**10.3.1.14 #define YYCOPY\_NEEDED 1**

**10.3.1.15 #define YYDEBUG 1**

**10.3.1.16 #define YYDPRINTF( Args )**

**Value:**

```
do {
    if (yydebug)
        YYFPRINTF Args;
} while (YYID (0))
```

10.3.1.17 #define YYEMPTY (-2)

10.3.1.18 #define YYEOF 0

10.3.1.19 #define YYERRCODE 256

10.3.1.20 #define yyerrok (yyerrstatus = 0)

10.3.1.21 #define YYERROR goto yyerrorlab

10.3.1.22 #define YYERROR\_VERBOSE

10.3.1.23 #define YYERROR\_VERBOSE 1

10.3.1.24 #define YYFAIL goto yyerrlab

10.3.1.25 #define YYFINAL 14

10.3.1.26 #define YYFPRINTF fprintf

10.3.1.27 #define YYFREE free

10.3.1.28 #define YYID( n )(n)

10.3.1.29 #define YYINITDEPTH 200

10.3.1.30 #define YYLAST 115

10.3.1.31 #define YYLEX yylex ()

10.3.1.32 #define YYLLOC\_DEFAULT( *Current*, *Rhs*, *N* )

**Value:**

```
do
    if (YYID (N))
    {
        (Current).first_line   = YRHSLOC (Rhs, 1).first_line;
        (Current).first_column = YRHSLOC (Rhs, 1).first_column;
        (Current).last_line    = YRHSLOC (Rhs, N).last_line;
```

```

        (Current).last_column = YRHSLOC (Rhs, N).last_column;    \
    }                                                            \
    else                                                         \
    {                                                            \
        (Current).first_line = (Current).last_line =           \
        YRHSLOC (Rhs, 0).last_line;                             \
        (Current).first_column = (Current).last_column =       \
        YRHSLOC (Rhs, 0).last_column;                           \
    }                                                            \
    while (YYID (0))

```

10.3.1.33 #define YYLSP\_NEEDED 0

10.3.1.34 #define YYMALLOC malloc

10.3.1.35 #define YYMAXDEPTH 10000

10.3.1.36 #define YYMAXUTOK 285

10.3.1.37 #define YYNNTS 39

10.3.1.38 #define YYNRULES 74

10.3.1.39 #define YYNSTATES 137

10.3.1.40 #define YYNTOKENS 31

10.3.1.41 #define YYPACT\_NINF -58

10.3.1.42 #define yypact\_value\_is\_default( *yystate* )((*yystate*) == (-58))

10.3.1.43 #define YYPOPSTACK( *N* )(*yyvsp* -= (*N*), *yyssp* -= (*N*))

10.3.1.44 #define YYPULL 1

10.3.1.45 #define YYPURE 0

10.3.1.46 #define YYPUSH 0

10.3.1.47 #define YYRECOVERING( ) (!*yyerrstatus*)

10.3.1.48 #define YRHSLOC( *Rhs*, *K* )((*Rhs*)[*K*])

10.3.1.49 #define YYSIZE\_MAXIMUM ((*YYSIZE\_T*)-1)

10.3.1.50 **#define YYSIZE\_T** unsigned int

10.3.1.51 **#define YYSKELETON\_NAME** "yacc.c"

10.3.1.52 **#define YYSTACK\_ALLOC** YYMALLOC

10.3.1.53 **#define YYSTACK\_ALLOC\_MAXIMUM** YYSIZE\_MAXIMUM

10.3.1.54 **#define YYSTACK\_BYTES( N )**

**Value:**

```
((N) * (sizeof (yytype_int16) + sizeof (YYSTYPE)) \
+ YYSTACK_GAP_MAXIMUM)
```

10.3.1.55 **#define YYSTACK\_FREE** YFREE

10.3.1.56 **#define YYSTACK\_GAP\_MAXIMUM** (sizeof (union yyalloc) - 1)

10.3.1.57 **#define YYSTACK\_RELOCATE( Stack\_alloc, Stack )**

**Value:**

```
do
{
    YYSIZE_T yynewbytes;
    YYCOPY (&yyvsptr->Stack_alloc, Stack, yysize);
    Stack = &yyvsptr->Stack_alloc;
    yynewbytes = YYSTACKsize * sizeof (*Stack) + YYSTACK_GAP_MAXIMUM;
   yyvsptr += yynewbytes / sizeof (*yyvsptr);
}
while (YYID (0))
```

10.3.1.58 **#define yystype** YYSTYPE

10.3.1.59 **#define YYSTYPE\_IS\_DECLARED** 1

10.3.1.60 **#define YYSTYPE\_IS\_TRIVIAL** 1

10.3.1.61 **#define YYSYNTAX\_ERROR**

**Value:**

```
yysyntax_error (&yymmsg_alloc, &yymmsg, \
                yyssp, ytoken)
```

10.3.1.62 #define YYTABLE\_NINF -39

10.3.1.63 #define yytable\_value\_is\_error( *yytable\_value* ) YYID (0)

10.3.1.64 #define YYTERROR 1

10.3.1.65 #define YYTOKEN\_TABLE 0

10.3.1.66 #define YYTOKENTYPE

10.3.1.67 #define YYTRANSLATE( *YYX* ) ((unsigned int) (YYX) <= YYMAXUTOK ?  
yytranslate[YYX] : YYUNDEFTOK)

10.3.1.68 #define YYUNDEFTOK 2

10.3.1.69 #define YYUSE( *e* )((void) (e))

### 10.3.2 Typedef Documentation

10.3.2.1 typedef union YYSTYPE YYSTYPE

10.3.2.2 typedef short int yytype\_int16

10.3.2.3 typedef short int yytype\_int8

10.3.2.4 typedef unsigned short int yytype\_uint16

10.3.2.5 typedef unsigned char yytype\_uint8

### 10.3.3 Enumeration Type Documentation

10.3.3.1 enum yytokentype

Enumerator:

***LBRACE***

***RBRACE***

***LBRACKET***

***RBRACKET***

***LPAREN***

***RPAREN***

***COMMA***

***COLON***

*LT*  
*LTE*  
*GT*  
*GTE*  
*SEMI*  
*OR*  
*UNION*  
*INVERSE*  
*EXISTS*  
*EQ*  
*ARROW*  
*ID*  
*INT*  
*INVALID\_ID*  
*DASH*  
*PLUS*  
*STAR*  
*UMINUS*  
*WAND*  
*AND*

### 10.3.4 Function Documentation

10.3.4.1 `static void yy_symbol_value_print ( yyoutput , yytype , yyvaluep )` [*static*]

10.3.4.2 `void yyerror ( const char * s )`

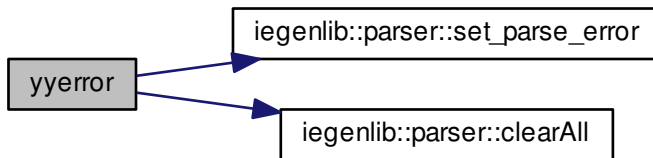
`yyerror` is called when an error occurs in parsing. It prints location of the error as well as why the error occurred to standard error. When an error occurs we set the `parse_error`;

#### Parameters

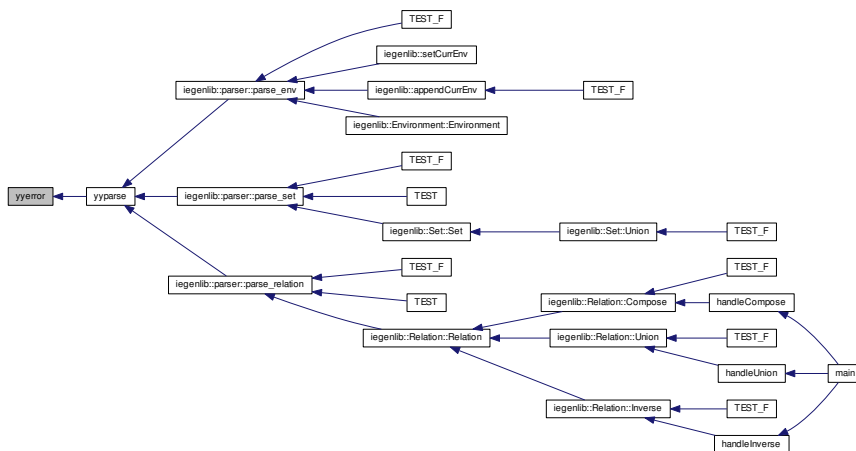
<i>const</i>	char pointer
--------------	--------------



Here is the call graph for this function:



Here is the caller graph for this function:



#### 10.3.4.3 int yyerror ( )



**10.3.5.1 int yychar****10.3.5.2 const yytype\_uint8 yycheck[]** [static]**Initial value:**

```
{
  15,  57,  59,  53,  60,  1,  7,  7,  4,  23,
  1,  1,  1,  3,  5,  5,  22,  6,  6,  15,
  9,  22,  22,  8,  39,  7,  41,  27,  43,  79,
  45,  81,  22,  29,  30,  92,  50,  7,  0,  95,
  25,  26,  6,  99, 100, 101, 102, 103, 104, 105,
  10,  3,  22,  23,  9,  25,  22,  23,  22,  23,
  8,  21,  5,  113, 1,  97,  98,  11,  12,  13,
  14,  8,  9,  20,  89,  90,  20,  17,  134, 129,
  21,  25,  26,  25,  26,  17,  15,  15,  3,  3,
  15,  18,  4,  4, 109,  4,  22,  10,  15,  21,
  7,  7,  21,  27,  7,  23,  4,  8,  28,  86,
  15,  27,  20,  89,  50,  15
}
```

**10.3.5.3 const yytype\_uint8 yydefact[]** [static]**Initial value:**

```
{
  0,  7,  0,  0,  0,  2,  3,  5,  4,  6,
  0,  28,  0,  0,  1,  0,  9,  13,  11,  16,
  0,  0,  0,  0,  0,  0,  40,  0,  0,  29,
  26,  0,  30,  37,  36,  0,  33,  35,  0,  0,
  0,  0,  21,  0,  14,  0,  17,  0,  32,  31,
  0,  0,  20,  40,  0,  23,  0,  0,  69,  59,
  0,  0,  0,  41,  44,  47,  48,  45,  46,  0,
  54,  56,  57,  58,  55,  66,  68,  67,  25,  40,
  15,  40,  0,  0,  34,  10,  0,  21,  12,  0,
  0,  0,  0,  63,  60,  0,  39,  0,  0,  0,
  0,  0,  0,  0,  0,  0,  0,  24,  0,  0,  0,
  0,  19,  22,  40,  70,  64,  74,  0,  43,  42,
  50,  51,  52,  53,  49,  62,  61,  65,  18,  40,
  8,  24,  72,  71,  0,  0,  73
}
```

**10.3.5.4 const yytype\_int8 yydefgoto[]** [static]**Initial value:**

```
{
  -1,  4,  5,  6,  7,  8,  9,  16,  17,  18,
  19,  51,  52,  54,  55,  10,  12,  53,  35,  36,
  61,  42,  62,  63,  64,  65,  66,  67,  68,  69,
  70,  71,  72,  73,  74,  75,  76,  77,  117
}
```

10.3.5.5 int yylineno = 1

10.3.5.6 YYSTYPE yylval

10.3.5.7 int yynerrs

10.3.5.8 const yytype\_int8 yypact[] [static]

Initial value:

```
{
  10,  -58,  -6,  18,  38,  -58,  -58,  -58,  -58,  -58,
  48,  45,  12,  52, -58,  57,  60,  -58,  68,  -58,
  -6,  59,  53,  36,  71,  72,  40,  85,  86,  -58,
 -58,  73, -58, -58, -58,  11, -58, -58,  57,  57,
  30,   9,  88,  57, -58,  57, -58,  74, -58, -58,
  34,  89,  75,  87,  91,  83,  78,  30,  93,   0,
  30,  94,   4, -58, -58, -58, -58, -58, -58,  56,
 -58, -58, -58, -58,  76, -58, -58, -58, -58,  87,
 -58,  87,  81,  97, -58, -58,  57, -58, -58,  57,
   9,  15,  -1, -58, -58,  30, -58,  30,  30,  30,
  30,  30,  30,  30,  30,  30,  82, 102,  88,  57,
  99, -58, -58,  87, -58, -58,  58,  63, -58, -58,
  58,  58,  58,  58,  58, -58, -58, -58, -58,  87,
 -58, -58, -58, -58,  30, 102,  58
}
```

10.3.5.9 const yytype\_int8 yypgoto[] [static]

Initial value:

```
{
  -58, -58, -58, -58, -58, -58, -58, -58,  84, -58,
  80,  23,  95,  24, 100, -58,  92, -15, -58,  64,
 -14, -50, -58, -32, -58, -58, -58, -58, -58, -56,
 -58, -58, -58, -58, -57, -58, -58, -58, -58
}
```

10.3.5.10 const yytype\_uint8 yyprhs[] [static]

Initial value:

```
{
  0,  0,  3,  5,  7,  9,  11,  13,  15,  24,
  27,  34,  37,  44,  46,  50,  55,  57,  61,  68,
  72,  74,  77,  81,  83,  88,  92,  97,  98, 100,
 104, 107, 111, 115, 117, 121, 123, 125, 127, 130,
 134, 135, 137, 141, 145, 147, 149, 151, 153, 155,
 159, 163, 167, 171, 175, 177, 179, 181, 183, 185,
}
```

```

    187, 190, 194, 198, 201, 205, 209, 211, 213, 215,
    217, 221, 226, 231, 235
}

```

#### 10.3.5.11 const yytype\_uint8 yyr1[] [static]

##### Initial value:

```

{
    0, 31, 32, 32, 32, 32, 32, 32, 33, 34,
    35, 36, 37, 38, 38, 39, 40, 40, 41, 42,
    42, 43, 44, 44, 45, 45, 46, 46, 47, 47,
    48, 48, 48, 49, 49, 50, 50, 51, 52, 52,
    52, 53, 53, 53, 54, 54, 54, 54, 54, 55,
    56, 57, 58, 59, 60, 60, 60, 60, 60, 61,
    62, 63, 63, 64, 64, 64, 65, 65, 65, 66,
    67, 68, 68, 69, 69
}

```

#### 10.3.5.12 const yytype\_uint8 yyr2[] [static]

##### Initial value:

```

{
    0, 2, 1, 1, 1, 1, 1, 1, 8, 2,
    6, 2, 6, 1, 3, 4, 1, 3, 6, 3,
    1, 2, 3, 1, 4, 3, 4, 0, 1, 3,
    2, 3, 3, 1, 3, 1, 1, 1, 2, 3,
    0, 1, 3, 3, 1, 1, 1, 1, 1, 3,
    3, 3, 3, 3, 1, 1, 1, 1, 1, 1,
    2, 3, 3, 2, 3, 3, 1, 1, 1, 1,
    3, 4, 4, 3, 1
}

```

#### 10.3.5.13 const yytype\_int8 yyrhs[] [static]

##### Initial value:

```

{
    32, 0, -1, 33, -1, 34, -1, 36, -1, 35,
    -1, 37, -1, 1, -1, 22, 7, 8, 20, 18,
    22, 7, 8, -1, 46, 38, -1, 46, 3, 43,
    15, 42, 4, -1, 46, 40, -1, 46, 3, 45,
    15, 44, 4, -1, 39, -1, 38, 17, 39, -1,
    3, 48, 52, 4, -1, 41, -1, 40, 17, 41,
    -1, 3, 48, 21, 48, 52, 4, -1, 43, 15,
    42, -1, 43, -1, 48, 52, -1, 45, 15, 44,
    -1, 45, -1, 48, 21, 48, 52, -1, 48, 21,
    1, -1, 5, 47, 6, 21, -1, -1, 22, -1,

```

```

22, 9, 47, -1, 5, 6, -1, 5, 49, 6,
-1, 5, 49, 1, -1, 50, -1, 49, 9, 50,
-1, 51, -1, 23, -1, 22, -1, 10, 53, -1,
10, 53, 1, -1, -1, 54, -1, 53, 30, 54,
-1, 53, 29, 54, -1, 55, -1, 58, -1, 59,
-1, 56, -1, 57, -1, 60, 20, 60, -1, 60,
11, 60, -1, 60, 12, 60, -1, 60, 13, 60,
-1, 60, 14, 60, -1, 61, -1, 65, -1, 62,
-1, 63, -1, 64, -1, 23, -1, 25, 60, -1,
60, 26, 60, -1, 60, 25, 60, -1, 23, 65,
-1, 23, 27, 65, -1, 65, 27, 23, -1, 66,
-1, 68, -1, 67, -1, 22, -1, 7, 60, 8,
-1, 51, 7, 69, 8, -1, 51, 7, 69, 1,
-1, 69, 9, 60, -1, 60, -1
}

```

#### 10.3.5.14 const yytype\_uint16 yrline[] [static]

##### Initial value:

```

{
    0, 152, 152, 155, 158, 161, 164, 167, 171, 179,
192, 209, 224, 241, 249, 255, 284, 292, 298, 339,
345, 353, 383, 389, 397, 438, 450, 454, 456, 464,
471, 474, 477, 488, 495, 501, 506, 510, 515, 518,
529, 532, 539, 545, 551, 554, 557, 560, 563, 566,
574, 583, 591, 600, 608, 611, 614, 617, 620, 624,
631, 637, 643, 650, 656, 662, 668, 671, 674, 677,
686, 689, 697, 707, 713
}

```

#### 10.3.5.15 int yrrule

#### 10.3.5.16 const char\* yysrc

#### 10.3.5.17 const yytype\_uint8 yystos[] [static]

##### Initial value:

```

{
    0, 1, 5, 22, 32, 33, 34, 35, 36, 37,
46, 22, 47, 7, 0, 3, 38, 39, 40, 41,
9, 6, 8, 5, 43, 45, 48, 17, 17, 47,
21, 20, 6, 22, 23, 49, 50, 51, 15, 15,
10, 21, 52, 3, 39, 3, 41, 18, 1, 6,
9, 42, 43, 48, 44, 45, 48, 7, 22, 23,
25, 51, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62, 63, 64, 65, 66, 67, 68, 1, 48,
4, 48, 48, 22, 50, 4, 15, 52, 4, 15,
21, 60, 27, 65, 60, 7, 1, 29, 30, 11,
}

```

```

    12,    13,    14,    20,    25,    26,    27,    52,    52,    21,
    7,     42,    44,    48,     8,    65,    60,    69,    54,    54,
    60,    60,    60,    60,    60,    60,    60,    23,     4,    48,
    8,     52,     1,     8,     9,    52,    60
}

```

### 10.3.5.18 const yytype\_int16 yytable[] [static]

#### Initial value:

```

{
    26,    91,    93,    87,    94,    96,    57,    57,   -38,    37,
    78,     1,    48,   -27,    23,     2,    11,    49,    21,   -38,
    50,    58,    58,   114,    56,    13,    79,    92,    81,   107,
    82,   108,     3,    97,    98,   115,    37,    57,    14,   116,
   104,   105,    32,   120,   121,   122,   123,   124,   125,   126,
    40,    15,    58,    59,    20,    60,    33,    34,    33,    34,
    22,    41,    23,   131,   132,   118,   119,    99,   100,   101,
   102,   133,   134,    31,    56,   113,   103,    27,   136,   135,
    30,   104,   105,   104,   105,    28,    38,    39,    43,    45,
    86,    47,    80,    85,   129,    88,    83,    40,    89,    90,
   -37,    95,   109,   106,   110,   127,   128,   130,    46,   111,
    24,    44,    29,   112,    84,    25
}

```

### 10.3.5.19 char \* yytext

### 10.3.5.20 const char\* const yytname[] [static]

#### Initial value:

```

{
    "$end", "error", "$undefined", "LBRACE", "RBRACE", "LBRACKET",
    "RBRACKET", "LPAREN", "RPAREN", "COMMA", "COLON", "LT", "LTE", "GT",
    "GTE", "SEMI", "OR", "UNION", "INVERSE", "EXISTS", "EQ", "ARROW", "ID",
    "INT", "INVALID_ID", "DASH", "PLUS", "STAR", "UMINUS", "WAND", "AND",
    "$accept", "Start", "environment", "omega_set", "isl_set",
    "omega_relation", "isl_relation", "conjunct_list", "single_conjunct",
    "rel_conjunct_list", "rel_single_conjunct", "semi_conjunct_list",
    "semi_conjunct", "semi_rel_conjunct_list", "semi_rel_conjunct",
    "in_symbolic", "symbol_list", "tuple", "tuple_elem_list", "tuple_elem",
    "variable_id", "optional_constraints", "constraint_list", "constraint",
    "constraint_eq", "constraint_lt", "constraint_lte", "constraint_gt",
    "constraint_gte", "expression", "expression_int", "expression_unop",
    "expression_binop", "expression_int_mult", "expression_simple",
    "expression_id", "expression_paren", "expression_func",
    "expression_list", 0
}

```

10.3.5.21 `yytype_int16*` `yytop`

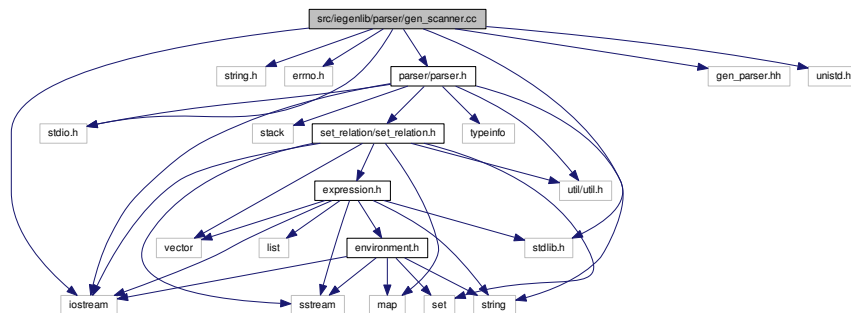
10.3.5.22 `const yytype_uint8 yytranslate[]` [`static`]

10.3.5.23 `int yytype`

10.3.5.24 `YYSTYPE *` `yyvaluep`

## 10.4 `src/iegenlib/parser/gen_scanner.cc` File Reference

```
#include <stdio.h> #include <string.h> #include <errno.-
h> #include <stdlib.h> #include <iostream> #include <parser/parser.-
h> #include "gen_parser.hh" #include <unistd.h> Include depen-
dency graph for gen_scanner.cc:
```



### Classes

- struct [yy\\_buffer\\_state](#)
- struct [yy\\_trans\\_info](#)

### Defines

- `#define YY_INT_ALIGNED` short int
- `#define FLEX_SCANNER`
- `#define YY_FLEX_MAJOR_VERSION` 2
- `#define YY_FLEX_MINOR_VERSION` 5
- `#define YY_FLEX_SUBMINOR_VERSION` 35
- `#define FLEX_BETA`



- #define [FLEXINT\\_H](#)
- #define [INT8\\_MIN](#) (-128)
- #define [INT16\\_MIN](#) (-32767-1)
- #define [INT32\\_MIN](#) (-2147483647-1)
- #define [INT8\\_MAX](#) (127)
- #define [INT16\\_MAX](#) (32767)
- #define [INT32\\_MAX](#) (2147483647)
- #define [UINT8\\_MAX](#) (255U)
- #define [UINT16\\_MAX](#) (65535U)
- #define [UINT32\\_MAX](#) (4294967295U)
- #define [yyconst](#)
- #define [YY\\_NULL](#) 0
- #define [YY\\_SC\\_TO\\_UI](#)(c) ((unsigned int) (unsigned char) c)
- #define [BEGIN](#) ([yy\\_start](#)) = 1 + 2 \*
- #define [YY\\_START](#) ((([yy\\_start](#)) - 1) / 2)
- #define [YYSTATE](#) [YY\\_START](#)
- #define [YY\\_STATE\\_EOF](#)(state) ([YY\\_END\\_OF\\_BUFFER](#) + state + 1)
- #define [YY\\_NEW\\_FILE](#) [yyrestart](#)([yyin](#) )
- #define [YY\\_END\\_OF\\_BUFFER\\_CHAR](#) 0
- #define [YY\\_BUF\\_SIZE](#) 16384
- #define [YY\\_STATE\\_BUF\\_SIZE](#) (([YY\\_BUF\\_SIZE](#) + 2) \* sizeof([yy\\_state\\_type](#)))
- #define [YY\\_TYPEDEF\\_Y\\_BUFFER\\_STATE](#)
- #define [EOB\\_ACT\\_CONTINUE\\_SCAN](#) 0
- #define [EOB\\_ACT\\_END\\_OF\\_FILE](#) 1
- #define [EOB\\_ACT\\_LAST\\_MATCH](#) 2
- #define [YY\\_LESS\\_LINENO](#)(n)
- #define [yyless](#)(n)
- #define [unput](#)(c) [yyunput](#)( c, ([yytext\\_ptr](#)) )
- #define [YY\\_TYPEDEF\\_Y\\_SIZE\\_T](#)
- #define [YY\\_STRUCT\\_Y\\_BUFFER\\_STATE](#)
- #define [YY\\_BUFFER\\_NEW](#) 0
- #define [YY\\_BUFFER\\_NORMAL](#) 1
- #define [YY\\_BUFFER\\_EOF\\_PENDING](#) 2
- #define [YY\\_CURRENT\\_BUFFER](#)
- #define [YY\\_CURRENT\\_BUFFER\\_LVALUE](#) ([yy\\_buffer\\_stack](#))[([yy\\_buffer\\_stack\\_top](#))]
- #define [YY\\_FLUSH\\_BUFFER](#) [yy\\_flush\\_buffer](#)([YY\\_CURRENT\\_BUFFER](#) )
- #define [yy\\_new\\_buffer](#) [yy\\_create\\_buffer](#)
- #define [yy\\_set\\_interactive](#)(is\_interactive)
- #define [yy\\_set\\_bol](#)(at\_bol)
- #define [YY\\_AT\\_BOL](#)() ([YY\\_CURRENT\\_BUFFER\\_LVALUE](#)->[yy\\_at\\_bol](#))
- #define [yytext\\_ptr](#) [yytext](#)
- #define [YY\\_DO\\_BEFORE\\_ACTION](#)

- #define `YY_NUM_RULES` 32
- #define `YY_END_OF_BUFFER` 33
- #define `REJECT` reject\_used\_but\_not\_detected
- #define `yymore()` yymore\_used\_but\_not\_detected
- #define `YY_MORE_ADJ` 0
- #define `YY_RESTORE_YY_MORE_OFFSET`
- #define `YY_NO_INPUT` 1
- #define `YY_INPUT`(buf, result, max\_size)
- #define `INITIAL` 0
- #define `YY_EXTRA_TYPE` void \*
- #define `YY_READ_BUF_SIZE` 8192
- #define `ECHO` do { if (fwrite( `ytext`, `yyleng`, 1, `yyout` )) {} } while (0)
- #define `yyterminate()` return YY\_NULL
- #define `YY_START_STACK_INCR` 25
- #define `YY_FATAL_ERROR`(msg) yy\_fatal\_error( msg )
- #define `YY_DECL_IS_OURS` 1
- #define `YY_DECL` int yylex (void)
- #define `YY_USER_ACTION`
- #define `YY_BREAK` break;
- #define `YY_RULE_SETUP` YY\_USER\_ACTION
- #define `YY_EXIT_FAILURE` 2
- #define `yylless`(n)
- #define `YYTABLES_NAME` "yytables"

## Typedefs

- typedef signed char `flex_int8_t`
- typedef short int `flex_int16_t`
- typedef int `flex_int32_t`
- typedef unsigned char `flex_uint8_t`
- typedef unsigned short int `flex_uint16_t`
- typedef unsigned int `flex_uint32_t`
- typedef struct `yy_buffer_state` \* `YY_BUFFER_STATE`
- typedef size\_t `yy_size_t`
- typedef unsigned char `YY_CHAR`
- typedef int `yy_state_type`

## Functions

- void `yyrestart` (FILE \*input\_file)
- void `yy_switch_to_buffer` (YY\_BUFFER\_STATE new\_buffer)
- YY\_BUFFER\_STATE `yy_create_buffer` (FILE \*file, int size)
- void `yy_delete_buffer` (YY\_BUFFER\_STATE b)
- void `yy_flush_buffer` (YY\_BUFFER\_STATE b)
- void `yypush_buffer_state` (YY\_BUFFER\_STATE new\_buffer)
- void `yypop_buffer_state` (void)
- static void `yyensure_buffer_stack` (void)
- static void `yy_load_buffer_state` (void)
- static void `yy_init_buffer` (YY\_BUFFER\_STATE b, FILE \*file)
- YY\_BUFFER\_STATE `yy_scan_buffer` (char \*base, yy\_size\_t size)
- YY\_BUFFER\_STATE `yy_scan_string` (yyconst char \*yy\_str)
- YY\_BUFFER\_STATE `yy_scan_bytes` (yyconst char \*bytes, int len)
- void \* `yyalloc` (yy\_size\_t)
- void \* `yyrealloc` (void \*, yy\_size\_t)
- void `yyfree` (void \*)
- static yy\_state\_type `yy_get_previous_state` (void)
- static yy\_state\_type `yy_try_NUL_trans` (yy\_state\_type current\_state)
- static int `yy_get_next_buffer` (void)
- static void `yy_fatal_error` (yyconst char msg[])
- static int `yy_init_globals` (void)
- int `yylex_destroy` (void)
- int `yyget_debug` (void)
- void `yyset_debug` (int debug\_flag)
- YY\_EXTRA\_TYPE `yyget_extra` (void)
- void `yyset_extra` (YY\_EXTRA\_TYPE user\_defined)
- FILE \* `yyget_in` (void)
- void `yyset_in` (FILE \*in\_str)
- FILE \* `yyget_out` (void)
- void `yyset_out` (FILE \*out\_str)
- int `yyget_leng` (void)
- char \* `yyget_text` (void)
- int `yyget_lineno` (void)
- void `yyset_lineno` (int line\_number)
- int `yywrap` (void)
- int `yylex` (void)
- static void `yy_fatal_error` (yyconst char \*msg)

## Variables

- int `yyleng`
- FILE \* `yyin` = (FILE \*) 0
- FILE \* `yyout` = (FILE \*) 0
- static size\_t `yy_buffer_stack_top` = 0
- static size\_t `yy_buffer_stack_max` = 0
- static YY\_BUFFER\_STATE \* `yy_buffer_stack` = 0
- static char `yy_hold_char`
- static int `yy_n_chars`
- static char \* `yy_c_buf_p` = (char \*) 0
- static int `yy_init` = 0
- static int `yy_start` = 0
- static int `yy_did_buffer_switch_on_eof`
- int `yylineno` = 1
- char \* `yytext`
- static yyconst flex\_int16\_t `yy_accept` [87]
- static yyconst flex\_int32\_t `yy_ec` [256]
- static yyconst flex\_int32\_t `yy_meta` [43]
- static yyconst flex\_int16\_t `yy_base` [97]
- static yyconst flex\_int16\_t `yy_def` [97]
- static yyconst flex\_int16\_t `yy_nxt` [272]
- static yyconst flex\_int16\_t `yy_chk` [272]
- static `yy_state_type` `yy_last_accepting_state`
- static char \* `yy_last_accepting_cpos`
- int `yy_flex_debug` = 0

### 10.4.1 Define Documentation

10.4.1.1 `#define BEGIN (yy_start) = 1 + 2 *`

10.4.1.2 `#define ECHO do { if (fwrite( yytext, yyleng, 1, yyout )) {} } while (0)`

10.4.1.3 `#define EOB_ACT_CONTINUE_SCAN 0`

10.4.1.4 `#define EOB_ACT_END_OF_FILE 1`

10.4.1.5 `#define EOB_ACT_LAST_MATCH 2`

10.4.1.6 `#define FLEX_BETA`

10.4.1.7 `#define FLEX_SCANNER`

- 10.4.1.8 #define FLEXINT\_H
- 10.4.1.9 #define INITIAL 0
- 10.4.1.10 #define INT16\_MAX (32767)
- 10.4.1.11 #define INT16\_MIN (-32767-1)
- 10.4.1.12 #define INT32\_MAX (2147483647)
- 10.4.1.13 #define INT32\_MIN (-2147483647-1)
- 10.4.1.14 #define INT8\_MAX (127)
- 10.4.1.15 #define INT8\_MIN (-128)
- 10.4.1.16 #define REJECT reject\_used\_but\_not\_detected
- 10.4.1.17 #define UINT16\_MAX (65535U)
- 10.4.1.18 #define UINT32\_MAX (4294967295U)
- 10.4.1.19 #define UINT8\_MAX (255U)
- 10.4.1.20 #define unput( c ) yyunput( c, (yytext\_ptr) )
- 10.4.1.21 #define YY\_AT\_BOL( ) (YY\_CURRENT\_BUFFER\_LVALUE->yy\_at\_bol)
- 10.4.1.22 #define YY\_BREAK break;
- 10.4.1.23 #define YY\_BUF\_SIZE 16384
- 10.4.1.24 #define YY\_BUFFER\_EOF\_PENDING 2
- 10.4.1.25 #define YY\_BUFFER\_NEW 0
- 10.4.1.26 #define YY\_BUFFER\_NORMAL 1
- 10.4.1.27 #define YY\_CURRENT\_BUFFER

**Value:**

```
( (yy_buffer_stack) \
    ? (yy_buffer_stack)[(yy_buffer_stack_top)] \
    : NULL)
```

10.4.1.28 `#define YY_CURRENT_BUFFER_LVALUE (yy_buffer_stack)[(yy_buffer_stack_top)]`

10.4.1.29 `#define YY_DECL int yylex (void)`

10.4.1.30 `#define YY_DECL_IS_OURS 1`

10.4.1.31 `#define YY_DO_BEFORE_ACTION`

**Value:**

```
(yytext_ptr) = yy_bp; \
  yyleng = (size_t) (yy_cp - yy_bp); \
  (yy_hold_char) = *yy_cp; \
  *yy_cp = '\0'; \
  (yy_c_buf_p) = yy_cp;
```

10.4.1.32 `#define YY_END_OF_BUFFER 33`

10.4.1.33 `#define YY_END_OF_BUFFER_CHAR 0`

10.4.1.34 `#define YY_EXIT_FAILURE 2`

10.4.1.35 `#define YY_EXTRA_TYPE void *`

10.4.1.36 `#define YY_FATAL_ERROR( msg ) yy_fatal_error(msg)`

10.4.1.37 `#define YY_FLEX_MAJOR_VERSION 2`

10.4.1.38 `#define YY_FLEX_MINOR_VERSION 5`

10.4.1.39 `#define YY_FLEX_SUBMINOR_VERSION 35`

10.4.1.40 `#define YY_FLUSH_BUFFER yy_flush_buffer(YY_CURRENT_BUFFER)`

10.4.1.41 `#define YY_INPUT( buf, result, max_size )`

**Value:**

```
{ \
  int c = iegenlib::parser::string_get_next_char(); \
  result = (c == EOF) ? YY_NULL : (buf[0] = c, 1); \
}
```

10.4.1.42 `#define YY_INT_ALIGNED` short int

10.4.1.43 `#define YY_LESS_LINENO( n )`

10.4.1.44 `#define YY_MORE_ADJ` 0

10.4.1.45 `#define yy_new_buffer` yy\_create\_buffer

10.4.1.46 `#define YY_NEW_FILE` yyrestart(yyin )

10.4.1.47 `#define YY_NO_INPUT` 1

10.4.1.48 `#define YY_NULL` 0

10.4.1.49 `#define YY_NUM_RULES` 32

10.4.1.50 `#define YY_READ_BUF_SIZE` 8192

10.4.1.51 `#define YY_RESTORE_YY_MORE_OFFSET`

10.4.1.52 `#define YY_RULE_SETUP` YY\_USER\_ACTION

10.4.1.53 `#define YY_SC_TO_UI( c )` ((unsigned int) (unsigned char) c)

10.4.1.54 `#define yy_set_bol( at_bol )`

**Value:**

```
{ \
    if ( ! YY_CURRENT_BUFFER ){ \
        yyensure_buffer_stack (); \
        YY_CURRENT_BUFFER_LVALUE = \
            yy_create_buffer(yyin,YY_BUF_SIZE ); \
    } \
    YY_CURRENT_BUFFER_LVALUE->yy_at_bol = at_bol; \
}
```

10.4.1.55 `#define yy_set_interactive( is_interactive )`

**Value:**

```
{ \
    if ( ! YY_CURRENT_BUFFER ){ \
        yyensure_buffer_stack (); \
        YY_CURRENT_BUFFER_LVALUE = \
            yy_create_buffer(yyin,YY_BUF_SIZE ); \
    } \
}
```

```
YY_CURRENT_BUFFER_LVALUE->yy_is_interactive = is_interactive; \
}
```

10.4.1.56 **#define YY\_START** (((yy\_start) - 1) / 2)

10.4.1.57 **#define YY\_START\_STACK\_INCR** 25

10.4.1.58 **#define YY\_STATE\_BUF\_SIZE** ((YY\_BUF\_SIZE + 2) \* sizeof(yy\_state\_type))

10.4.1.59 **#define YY\_STATE\_EOF**( state ) (YY\_END\_OF\_BUFFER + state + 1)

10.4.1.60 **#define YY\_STRUCT\_YY\_BUFFER\_STATE**

10.4.1.61 **#define YY\_TYPEDEF\_YY\_BUFFER\_STATE**

10.4.1.62 **#define YY\_TYPEDEF\_YY\_SIZE\_T**

10.4.1.63 **#define YY\_USER\_ACTION**

10.4.1.64 **#define yyconst**

10.4.1.65 **#define yyless**( n )

**Value:**

```
do \
    { \
        /* Undo effects of setting up yytext. */ \
        int yyless_macro_arg = (n); \
        YY_LESS_LINENO(yyless_macro_arg); \
        *yy_cp = (yy_hold_char); \
        YY_RESTORE_YY_MORE_OFFSET \
        (yy_c_buf_p) = yy_cp = yy_bp + yyless_macro_arg - YY_MORE_ADJ; \
        YY_DO_BEFORE_ACTION; /* set up yytext again */ \
    } \
    while ( 0 )
```

10.4.1.66 **#define yyless**( n )

**Value:**

```
do \
    { \
        /* Undo effects of setting up yytext. */ \
        int yyless_macro_arg = (n); \
```



```
YY_LESS_LINENO(yyles_macro_arg);\n    yytext[yyval] = (yy_hold_char); \n    (yy_c_buf_p) = yytext + yyles_macro_arg; \n    (yy_hold_char) = *(yy_c_buf_p); \n    *(yy_c_buf_p) = '\\0'; \n    yyval = yyles_macro_arg; \n    } \nwhile ( 0 )
```

10.4.1.67 **#define yymore( ) yymore\_used\_but\_not\_detected**

10.4.1.68 **#define YYSTATE YY\_START**

10.4.1.69 **#define YYTABLES\_NAME "yytables"**

10.4.1.70 **#define yyterminate( ) return YY\_NULL**

10.4.1.71 **#define yytext\_ptr yytext**

## 10.4.2 Typedef Documentation

10.4.2.1 **typedef short int flex\_int16\_t**

10.4.2.2 **typedef int flex\_int32\_t**

10.4.2.3 **typedef signed char flex\_int8\_t**

10.4.2.4 **typedef unsigned short int flex\_uint16\_t**

10.4.2.5 **typedef unsigned int flex\_uint32\_t**

10.4.2.6 **typedef unsigned char flex\_uint8\_t**

10.4.2.7 **typedef struct yy\_buffer\_state\* YY\_BUFFER\_STATE**

10.4.2.8 **typedef unsigned char YY\_CHAR**

10.4.2.9 **typedef size\_t yy\_size\_t**

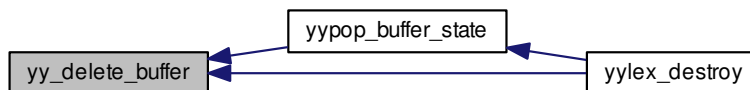
10.4.2.10 **typedef int yy\_state\_type**

## 10.4.3 Function Documentation

10.4.3.1 **YY\_BUFFER\_STATE yy\_create\_buffer ( FILE \* file, int size )**

10.4.3.2 `void yy_delete_buffer ( YY_BUFFER_STATE b )`

Here is the caller graph for this function:



10.4.3.3 `static void yy_fatal_error ( yyconst char msg[] )` [static]

10.4.3.4 `static void yy_fatal_error ( yyconst char * msg )` [static]

10.4.3.5 `void yy_flush_buffer ( YY_BUFFER_STATE b )`

Discard all buffered characters. On the next scan, `YY_INPUT` will be called.

#### Parameters

<i>b</i>	the buffer state to be flushed, usually <code>YY_CURRENT_BUFFER</code> .
----------	--

Here is the call graph for this function:



Here is the caller graph for this function:



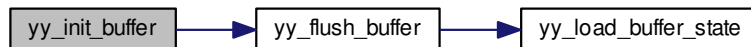
10.4.3.6 `static int yy_get_next_buffer ( void ) [static]`

10.4.3.7 `static yy_state_type yy_get_previous_state ( void ) [static]`

10.4.3.8 `static void yy_init_buffer ( YY_BUFFER_STATE b, FILE * file ) [static]`

The main scanner function which does all the work.

Here is the call graph for this function:



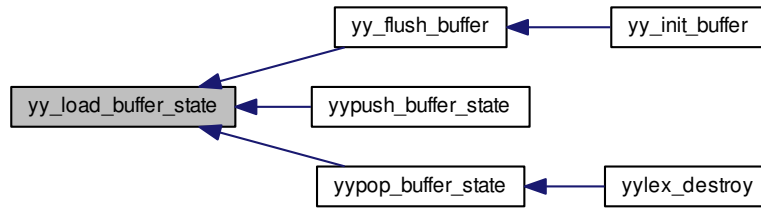
#### 10.4.3.9 static int yy\_init\_globals ( void ) [static]

Here is the caller graph for this function:



#### 10.4.3.10 static void yy\_load\_buffer\_state ( void ) [static]

Here is the caller graph for this function:



#### 10.4.3.11 YY\_BUFFER\_STATE yy\_scan\_buffer ( char \* base, yy\_size\_t size )

Setup the input buffer state to scan directly from a user-specified character buffer.

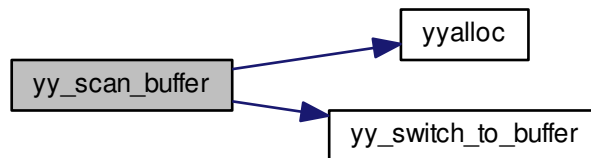
##### Parameters

<i>base</i>	the character buffer
<i>size</i>	the size in bytes of the character buffer

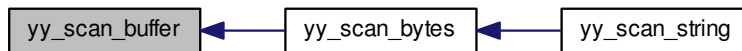
**Returns**

the newly allocated buffer state object.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 10.4.3.12 YY\_BUFFER\_STATE yy\_scan\_bytes ( yyconst char \* yybytes, int yybytes.len )

Setup the input buffer state to scan the given bytes. The next call to `yylex()` will scan from a *copy* of *bytes*.

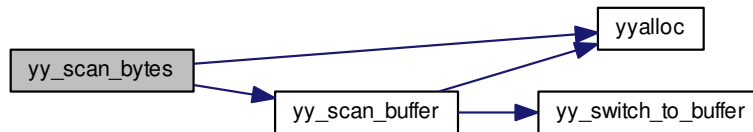
**Parameters**

<i>bytes</i>	the byte buffer to scan
<i>len</i>	the number of bytes in the buffer pointed to by <i>bytes</i> .

**Returns**

the newly allocated buffer state object.

Here is the call graph for this function:



Here is the caller graph for this function:



#### 10.4.3.13 YY\_BUFFER\_STATE yy\_scan\_string ( *yyconst* char \* *yyst* )

Setup the input buffer state to scan a string. The next call to `yylex()` will scan from a *copy* of *str*.

**Parameters**

<i>yyst</i>	a NUL-terminated string to scan
-------------	---------------------------------

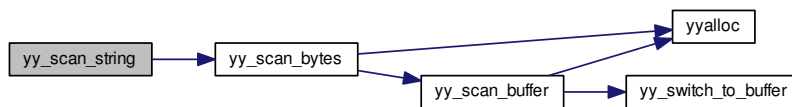
**Returns**

the newly allocated buffer state object.

**Note**

If you want to scan bytes that may contain NUL values, then use [yy\\_scan\\_bytes\(\)](#) instead.

Here is the call graph for this function:

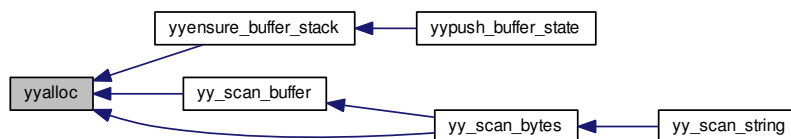
**10.4.3.14 void yy\_switch\_to\_buffer ( YY\_BUFFER\_STATE new\_buffer )**

Here is the caller graph for this function:

**10.4.3.15 static yy\_state\_type yy\_try\_NUL\_trans ( yy\_state\_type current\_state )**  
[static]

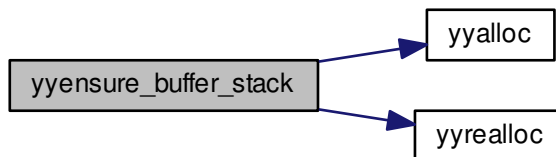
#### 10.4.3.16 void \* yyalloc ( yy\_size\_t size )

Here is the caller graph for this function:



#### 10.4.3.17 static void yyensure\_buffer\_stack ( void ) [static]

Here is the call graph for this function:



Here is the caller graph for this function:





## 10.4.3.18 void yyfree ( void \* ptr )

Here is the caller graph for this function:



## 10.4.3.19 int yyget\_debug ( void )

## 10.4.3.20 YY\_EXTRA\_TYPE yyget\_extra ( void )

## 10.4.3.21 FILE \* yyget\_in ( void )

Get the input stream.

## 10.4.3.22 int yyget\_leng ( void )

Get the length of the current token.

## 10.4.3.23 int yyget\_lineno ( void )

Get the current line number.

## 10.4.3.24 FILE \* yyget\_out ( void )

Get the output stream.

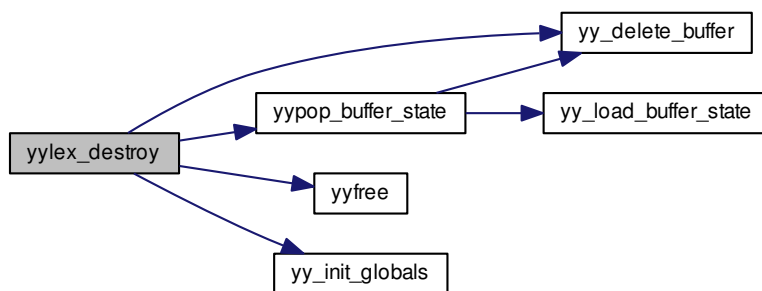
## 10.4.3.25 char \* yyget\_text ( void )

Get the current token.

10.4.3.26 `int yylex ( void )`

10.4.3.27 `int yylex_destroy ( void )`

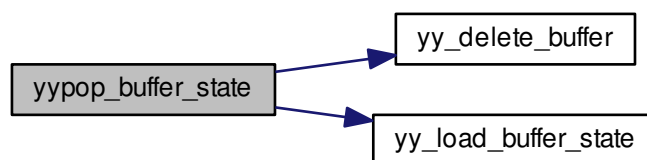
Here is the call graph for this function:



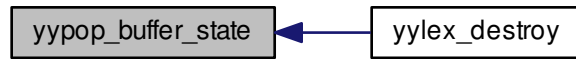
10.4.3.28 `void yypop_buffer_state ( void )`

Removes and deletes the top of the stack, if present. The next element becomes the new top.

Here is the call graph for this function:



Here is the caller graph for this function:



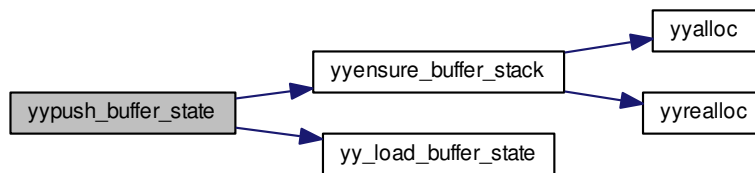
#### 10.4.3.29 void yypush\_buffer\_state ( YY\_BUFFER\_STATE *new\_buffer* )

Pushes the new state onto the stack. The new state becomes the current state. This function will allocate the stack if necessary.

##### Parameters

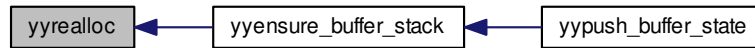
<i>new_buffer</i>	The new state.
-------------------	----------------

Here is the call graph for this function:



10.4.3.30 `void * yyrealloc ( void * ptr, yy_size_t size )`

Here is the caller graph for this function:



10.4.3.31 `void yyrestart ( FILE * input_file )`

10.4.3.32 `void yyset_debug ( int debug_flag )`

10.4.3.33 `void yyset_extra ( YY_EXTRA_TYPE user_defined )`

10.4.3.34 `void yyset_in ( FILE * in_str )`

Set the input stream. This does not discard the current input buffer.

#### Parameters

<i>in_str</i>	A readable stream.
---------------	--------------------

#### See also

[yy\\_switch\\_to\\_buffer](#)

10.4.3.35 `void yyset_lineno ( int line_number )`

Set the current line number.

#### Parameters

<i>line_number</i>	
--------------------	--

10.4.3.36 `void yyset_out ( FILE * out_str )`

10.4.3.37 int yywrap ( void )

## 10.4.4 Variable Documentation

10.4.4.1 yyconst flex\_int16\_t yy\_accept[87] [static]

**Initial value:**

```
{
  0,
  0, 0, 33, 32, 31, 28, 32, 28, 8, 9,
  3, 1, 10, 2, 29, 11, 19, 13, 12, 14,
  28, 28, 28, 28, 6, 7, 32, 28, 28, 28,
  28, 4, 5, 28, 28, 28, 28, 18, 28, 17,
  29, 15, 16, 28, 28, 22, 28, 28, 28, 0,
  28, 28, 23, 28, 28, 28, 27, 28, 28, 30,
  30, 28, 28, 28, 28, 28, 28, 30, 30, 30,
  30, 30, 28, 28, 28, 28, 20, 28, 28, 21,
  28, 24, 28, 26, 25, 0
}
```

10.4.4.2 yyconst flex\_int16\_t yy\_base[97] [static]

**Initial value:**

```
{
  0,
  0, 0, 174, 229, 229, 40, 166, 32, 229, 229,
  229, 229, 229, 141, 126, 229, 229, 113, 229, 78,
  33, 34, 35, 38, 229, 229, 45, 37, 39, 41,
  49, 229, 229, 71, 61, 0, 75, 229, 68, 229,
  46, 229, 229, 69, 73, 72, 74, 94, 77, 87,
  80, 81, 84, 96, 106, 101, 103, 107, 109, 113,
  110, 111, 118, 122, 123, 130, 133, 147, 132, 0,
  154, 134, 139, 153, 149, 155, 156, 159, 160, 161,
  180, 163, 175, 169, 182, 229, 211, 213, 55, 215,
  217, 52, 219, 221, 223, 225
}
```

10.4.4.3 YY\_BUFFER\_STATE\* yy\_buffer\_stack = 0 [static]

Stack as an array.

10.4.4.4 size\_t yy\_buffer\_stack\_max = 0 [static]

capacity of stack.

10.4.4.5 `size_t yy_buffer_stack_top = 0` [static]

index of top of stack.

10.4.4.6 `char* yy_c_buf_p = (char *)0` [static]

10.4.4.7 `yyconst flex_int16_t yy_chk[272]` [static]

10.4.4.8 `yyconst flex_int16_t yy_def[97]` [static]

**Initial value:**

```
{
  0,
  86,  1,  86,  86,  86,  87,  86,  88,  86,  86,
  86,  86,  86,  86,  86,  86,  86,  86,  86,  86,
  88,  88,  88,  88,  86,  86,  89,  88,  88,  88,
  88,  86,  86,  87,  88,  90,  91,  86,  88,  86,
  86,  86,  86,  88,  88,  88,  88,  88,  87,  88,  92,
  88,  88,  88,  88,  91,  93,  88,  88,  88,  94,
  95,  88,  88,  88,  93,  88,  88,  94,  95,  96,
  94,  95,  88,  88,  88,  88,  88,  88,  88,  88,
  88,  88,  88,  88,  88,  0,  86,  86,  86,  86,
  86,  86,  86,  86,  86,  86
}
```

10.4.4.9 `int yy_did_buffer_switch_on_eof` [static]

10.4.4.10 `yyconst flex_int32_t yy_ec[256]` [static]

10.4.4.11 `int yy_flex_debug = 0`

10.4.4.12 `char yy_hold_char` [static]

10.4.4.13 `int yy_init = 0` [static]

10.4.4.14 `char* yy_last_accepting_cpos` [static]

10.4.4.15 `yy_state_type yy_last_accepting_state` [static]

10.4.4.16 `yyconst flex_int32_t yy_meta[43]` [static]

**Initial value:**

```
{
  0,
  1,  1,  2,  1,  2,  1,  1,  1,  1,  1,

```

```

    1,  3,  1,  1,  1,  1,  1,  2,  2,  2,
    2,  2,  2,  2,  2,  2,  2,  1,  1,  2,
    2,  2,  2,  2,  2,  2,  2,  2,  2,  2,
    1,  1
}

```

10.4.4.17 `int yy_n_chars` [static]

10.4.4.18 `yyconst flex_int16_t yy_nxt[272]` [static]

10.4.4.19 `int yy_start = 0` [static]

10.4.4.20 `FILE * yyin = (FILE *) 0`

10.4.4.21 `int yyleng`

10.4.4.22 `int yylineno = 1`

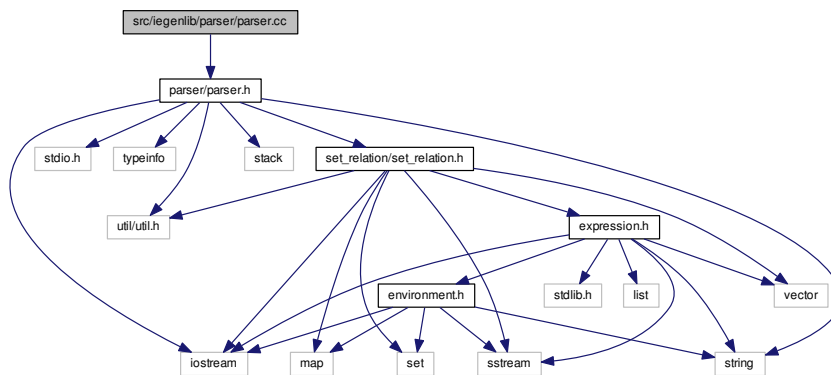
10.4.4.23 `FILE * yyout = (FILE *) 0`

10.4.4.24 `char* yytext`

## 10.5 src/iegenlib/parser/parser.cc File Reference

Implementations to methods declared in [parser.h](#).

`#include <parser/parser.h>` Include dependency graph for parser.cc:



## Namespaces

- namespace [iegenlib](#)
- namespace [iegenlib::parser](#)

## Functions

- int [yyparse](#) ()
- int [iegenlib::parser::string\\_get\\_next\\_char](#) ()  
*Used by lexer to obtain each character of input string being parsed.*
- Environment \* [iegenlib::parser::parse\\_env](#) (std::string env\_string)
- Environment \* [iegenlib::parser::get\\_parse\\_env\\_result](#) ()
- void [iegenlib::parser::set\\_parse\\_env\\_result](#) (Environment \*e)
- Set \* [iegenlib::parser::parse\\_set](#) (string set\_string)
- Set [iegenlib::parser::get\\_parse\\_set\\_result](#) ()
- void [iegenlib::parser::set\\_parse\\_set\\_result](#) (Set \*s)
- Relation \* [iegenlib::parser::parse\\_relation](#) (string relation\_string)
- Relation [iegenlib::parser::get\\_parse\\_relation\\_result](#) ()
- void [iegenlib::parser::set\\_parse\\_relation\\_result](#) (Relation \*s)
- void [iegenlib::parser::set\\_parse\\_error](#) (string error)
- bool [iegenlib::parser::get\\_parse\\_error](#) ()
- void [iegenlib::parser::clearAll](#) ()

## Variables

- string [iegenlib::parser::input\\_buffer](#)
- unsigned int [iegenlib::parser::input\\_pos](#)
- bool [iegenlib::parser::parse\\_error](#)
- string [iegenlib::parser::error\\_message](#)  
*error message for when an error occurs in the parser*
- Environment \* [iegenlib::parser::parse\\_env\\_result](#)
- Relation \* [iegenlib::parser::parse\\_relation\\_result](#)
- Set \* [iegenlib::parser::parse\\_set\\_result](#)

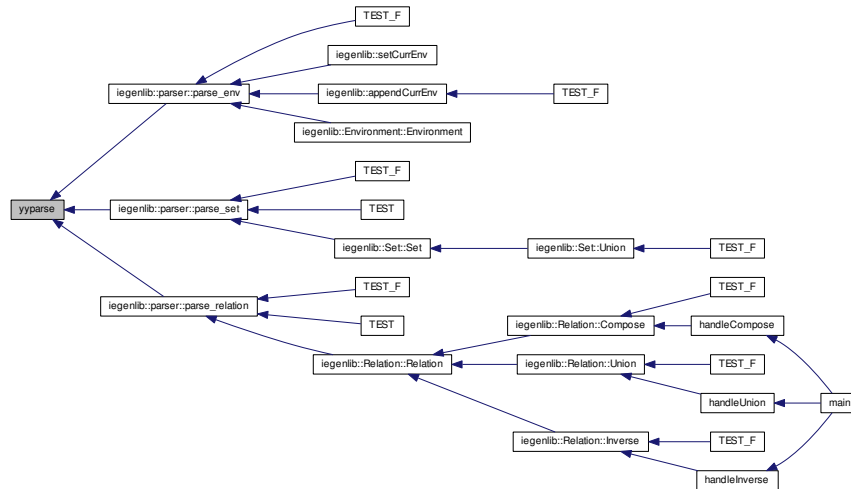
### 10.5.1 Detailed Description

Implementations to methods declared in [parser.h](#).





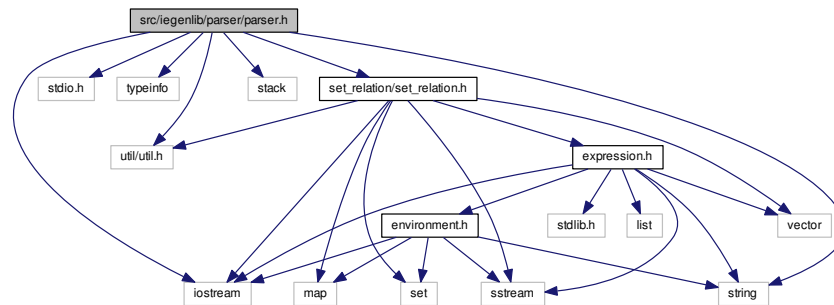
Here is the caller graph for this function:



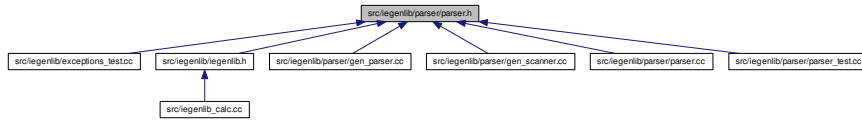
## 10.6 src/iegenlib/parser/parser.h File Reference

a namespace to allow the user to make use of the parser

```
#include <iostream> #include <stdio.h> #include <typeinfo> ×
#include <string> #include <stack> #include <set_relation/set-
_relation.h> #include <util/util.h> Include dependency graph for
parser.h:
```



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



## Namespaces

- namespace `iegenlib`
- namespace `iegenlib::parser`

## Functions

- `int iegenlib::parser::string_get_next_char ()`  
*Used by lexer to obtain each character of input string being parsed.*
- `Environment * iegenlib::parser::parse_env (std::string env_string)`
- `void iegenlib::parser::set_parse_env_result (Environment *e)`
- `Set * iegenlib::parser::parse_set (std::string set_string)`
- `void iegenlib::parser::set_parse_set_result (Set *s)`
- `Relation * iegenlib::parser::parse_relation (std::string relation_string)`
- `void iegenlib::parser::set_parse_relation_result (Relation *s)`
- `void iegenlib::parser::set_parse_error (std::string err)`
- `bool iegenlib::parser::get_parse_error ()`
- `void iegenlib::parser::clearAll ()`

### 10.6.1 Detailed Description

a namespace to allow the user to make use of the parser

#### Date

Started: 5/17/2010 #

#### Revision:

457

: last committed revision #

**Date:**

2012-06-28 13:41:00 -0#

: date of last committed revision #

**Author:**

mstrout

: author of last committed revision

**Authors**

Michelle Strout, Alan LaMielle, Nicholas Jeanette

Copyright (c) 2009, 2010, 2011, 2012, Colorado State University

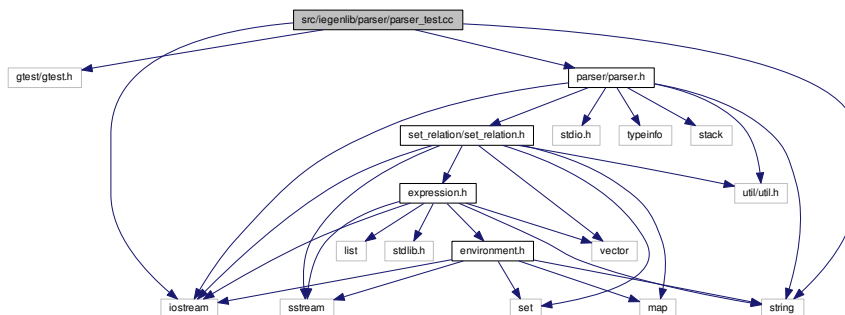
All rights reserved.

See ../COPYING for details.

## 10.7 src/iegenlib/parser/parser\_test.cc File Reference

Test file for parsing sets and relations from an omega-like syntax.

```
#include <gtest/gtest.h> #include <string> #include <parser/parser.h> #include <iostream> Include dependency graph for parser_test.cc:
```



### Functions

- [TEST](#) (Parser, OmegaSetEmpty)

- [TEST](#) (Parser, OmegaSet1Tuple)
- [TEST](#) (Parser, OmegaSet2Tuple)
- [TEST](#) (Parser, OmegaSetDuplicateTupleVar)
- [TEST](#) (Parser, OmegaSetUnion)
- [TEST](#) (Parser, OmegaSetInEqualityConstraintGT)
- [TEST](#) (Parser, OmegaSetInEqualityConstraintGTE)
- [TEST](#) (Parser, OmegaSetInEqualityConstraintLT)
- [TEST](#) (Parser, OmegaSetInEqualityConstraintLTE)
- [TEST](#) (Parser, OmegaSetEqualityConstraintEQ)
- [TEST](#) (Parser, OmegaRelation3to3)
- [TEST](#) (Parser, OmegaRelation1to3)
- [TEST](#) (Parser, OmegaRelationUnion)
- [TEST](#) (Parser, OmegaRelationInEqualityConstraintGT)
- [TEST](#) (Parser, OmegaRelationInEqualityConstraintGTE)
- [TEST](#) (Parser, OmegaRelationInEqualityConstraintLT)
- [TEST](#) (Parser, OmegaRelationInEqualityConstraintLTE)
- [TEST](#) (Parser, OmegaRelationEqualityConstraintEQ)
- [TEST](#) (Parser, SetNoConstraints1)
- [TEST](#) (Parser, SetNoConstraints2)
- [TEST](#) (Parser, RelationNoConstraints1)
- [TEST](#) (Parser, RelationNoConstraints2)
- [TEST](#) (Parser, SetConstraintsGT)
- [TEST](#) (Parser, RelationConstraintsEQ)
- [TEST](#) (Parser, RelationConstraintsLT)
- [TEST](#) (Parser, SetConstraintsGTE)
- [TEST](#) (Parser, RelationConstraintsGTE)
- [TEST](#) (Parser, RelationConstraintsEQandEQandLTE)
- [TEST](#) (Parser, RelationConstraintsLTExp)
- [TEST](#) (Parser, RelationConstraintsExpGTEandGTE)
- [TEST](#) (Parser, SetConstraintsExpGTEandExpGTE)
- [TEST](#) (Parser, SetDuplicateTupleVar)
- [TEST](#) (Parser, RelationDuplicateTupleVar)
- [TEST](#) (Parser, SetDuplicateTupleVarConstraintsEXPEQ)
- [TEST](#) (Parser, SetUndeclaredSymbolic)
- [TEST](#) (Parser, SetUndeclaredSymbolicConstraintLTandLT)
- [TEST](#) (Parser, SetUndeclaredSymbolicDuplicateTupleVarConstraintLTandLT)
- [TEST](#) (Parser, SetUndeclaredSymbolicConstraintExpLT)
- [TEST](#) (Parser, SetUndeclaredSymbolicConstraintGT)
- [TEST](#) (Parser, SetUndeclaredSymbolicConstraintGT2)
- [TEST](#) (Parser, SetUFCallConstraintLT)
- [TEST](#) (Parser, SetUFCallConstraintExpGTE)
- [TEST](#) (Parser, RelationComposeR1)
- [TEST](#) (Parser, RelationComposeR2)

- [TEST](#) (Parser, RelationComposeExpected)
- [TEST](#) (Parser, RestrictedIdent)
- [TEST](#) (Parser, OmegaSetExpMult)
- [TEST](#) (Parser, ISLSetUnion)
- [TEST](#) (Parser, ISLSetUnion2)
- [TEST](#) (Parser, OmegaSetInSymbolic)
- [TEST](#) (Parser, OmegaSetInSymbolicUnion)
- [TEST](#) (Parser, ISLSetInSymbolicUnion)
- [TEST](#) (Parser, ISLRelationUnion1)
- [TEST](#) (Parser, ISLRelationInSymbolicUnion1)
- [TEST](#) (Parser, ErrorHandlingBug)

### 10.7.1 Detailed Description

Test file for parsing sets and relations from an omega-like syntax.

#### Date

Started: 5/17/2010 #

#### Revision:

496

: last committed revision #

#### Date:

2012-08-22 22:30:57 -0#

: date of last committed revision #

#### Author:

mstrout

: author of last committed revision

#### Authors

Michelle Strout, Alan LaMielle, Nicholas Jeanette

Copyright (c) 2009, 2010, 2011, 2012, Colorado State University

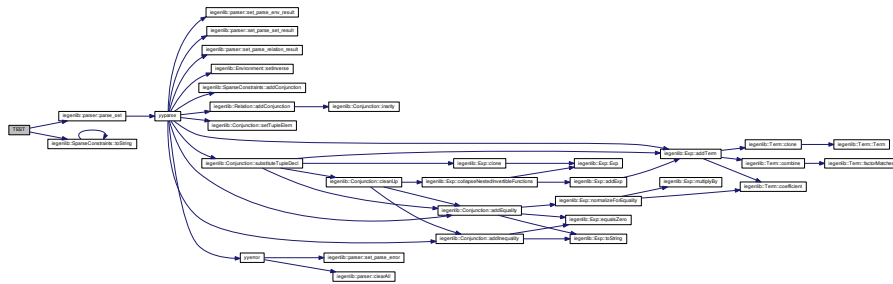
All rights reserved.

See ../COPYING for details.

## 10.7.2 Function Documentation

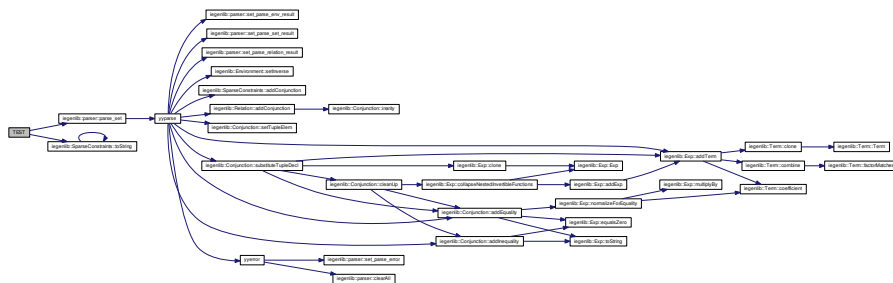
### 10.7.2.1 TEST ( Parser , OmegaSetEmpty )

Here is the call graph for this function:



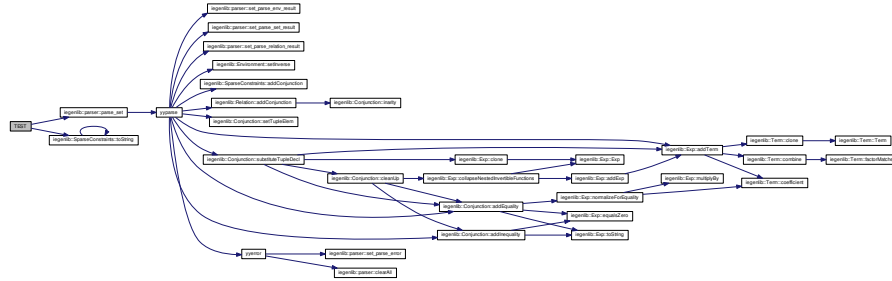
### 10.7.2.2 TEST ( Parser , OmegaSet1Tuple )

Here is the call graph for this function:



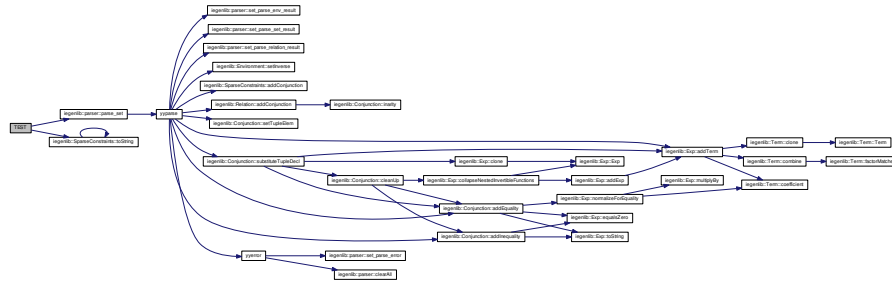
10.7.2.3 TEST ( Parser , OmegaSet2Tuple )

Here is the call graph for this function:



10.7.2.4 TEST ( Parser , OmegaSetDuplicateTupleVar )

Here is the call graph for this function:









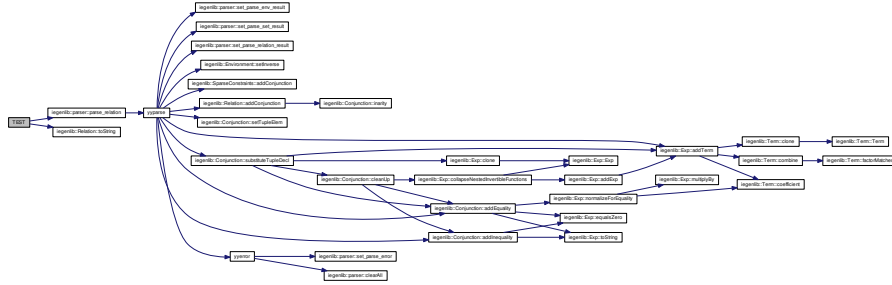






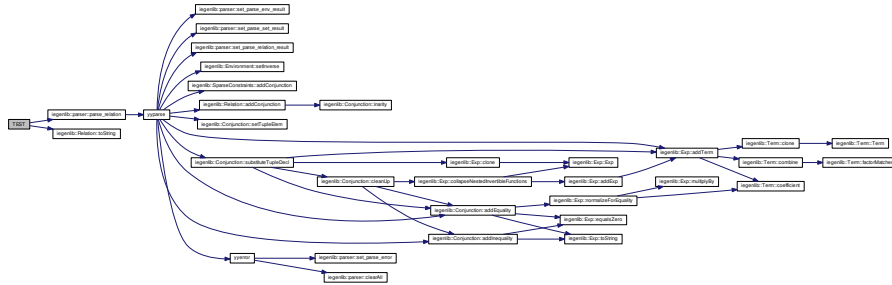
10.7.2.15 TEST ( Parser , OmegaRelationInEqualityConstraintGTE )

Here is the call graph for this function:



10.7.2.16 TEST ( Parser , OmegaRelationInEqualityConstraintLT )

Here is the call graph for this function:



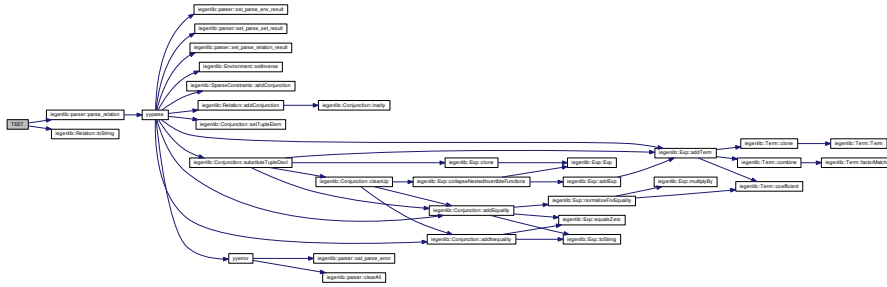






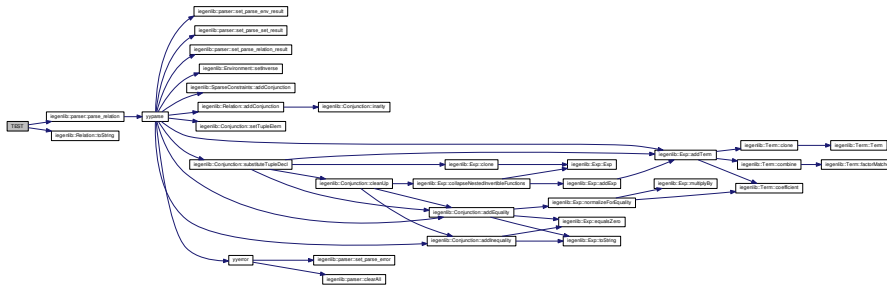
10.7.2.21 TEST ( Parser , RelationNoConstraints1 )

Here is the call graph for this function:



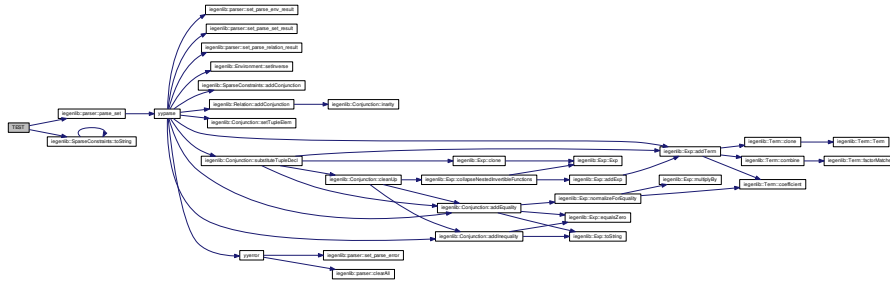
10.7.2.22 TEST ( Parser , RelationNoConstraints2 )

Here is the call graph for this function:



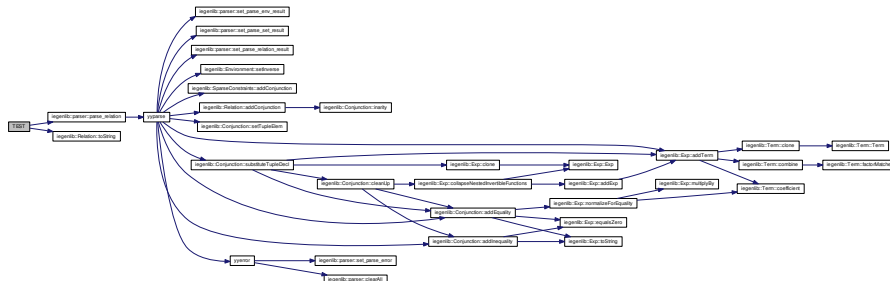
10.7.2.23 TEST ( Parser , SetConstraintsGT )

Here is the call graph for this function:



10.7.2.24 TEST ( Parser , RelationConstraintsEQ )

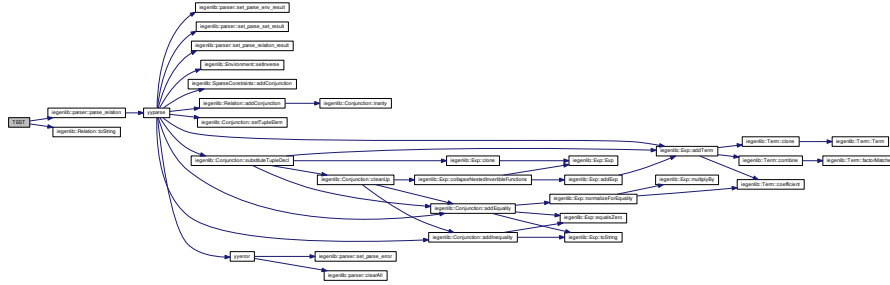
Here is the call graph for this function:





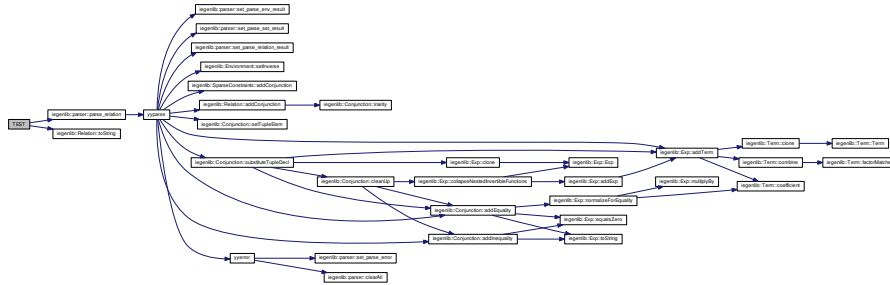
10.7.2.27 TEST ( Parser , RelationConstraintsGTE )

Here is the call graph for this function:



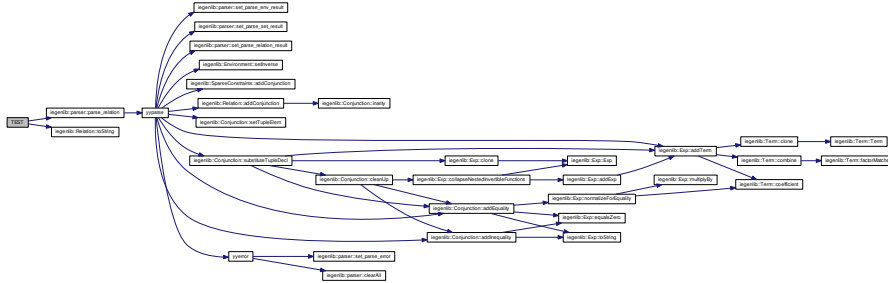
10.7.2.28 TEST ( Parser , RelationConstraintsEQandEQandLTE )

Here is the call graph for this function:



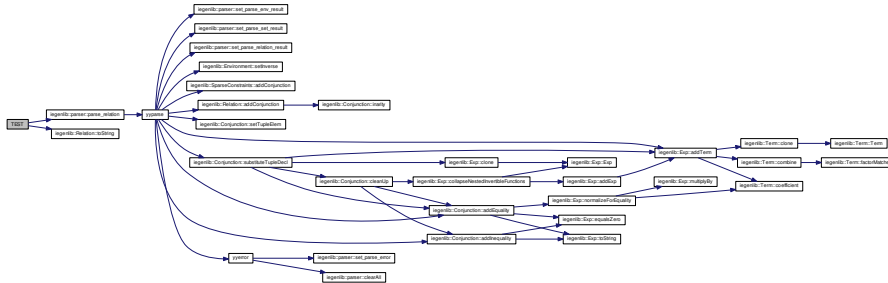
10.7.2.29 TEST ( Parser , RelationConstraintsLExp )

Here is the call graph for this function:



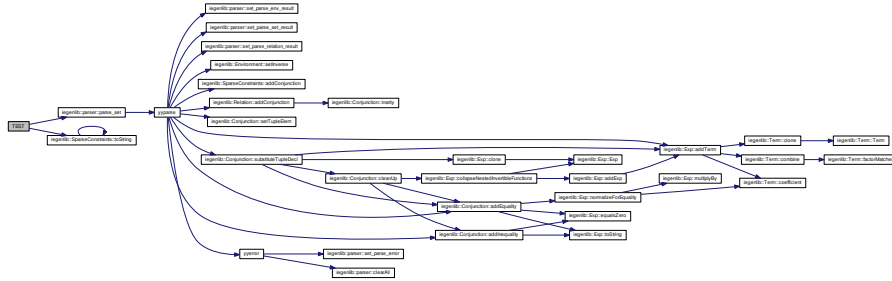
10.7.2.30 TEST ( Parser , RelationConstraintsExpGTEandGTE )

Here is the call graph for this function:



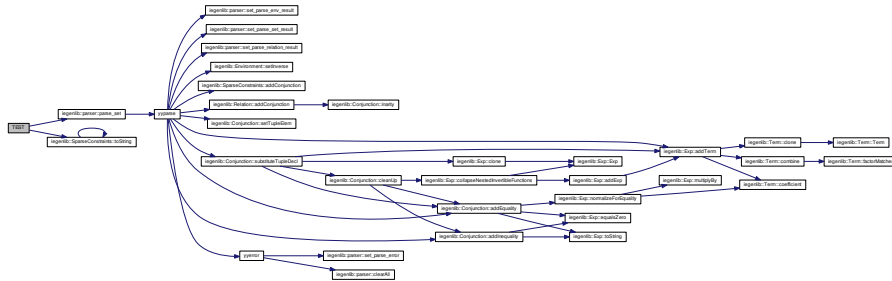
10.7.2.31 TEST ( Parser , SetConstraintsExpGTEandExpGTE )

Here is the call graph for this function:



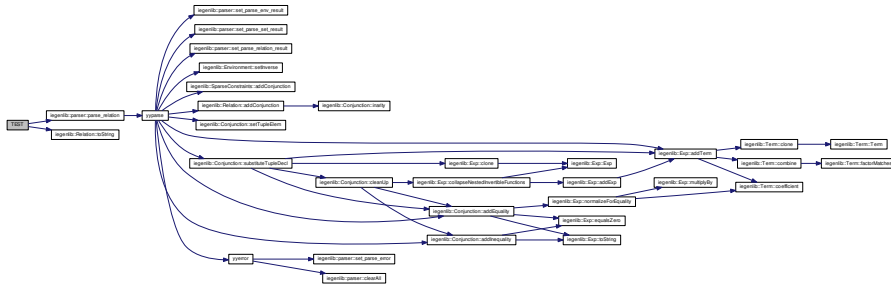
10.7.2.32 TEST ( Parser , SetDuplicateTupleVar )

Here is the call graph for this function:



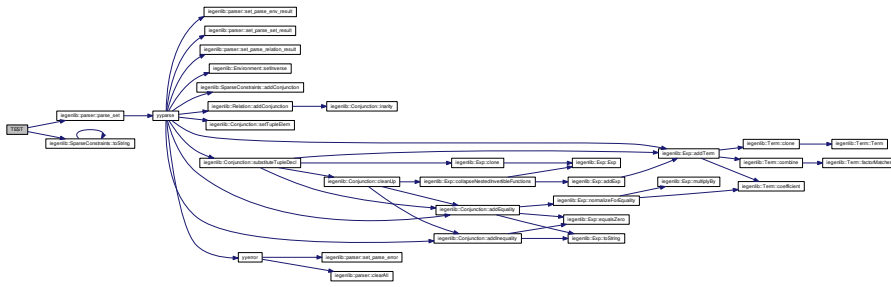
10.7.2.33 TEST ( Parser , RelationDuplicateTupleVar )

Here is the call graph for this function:



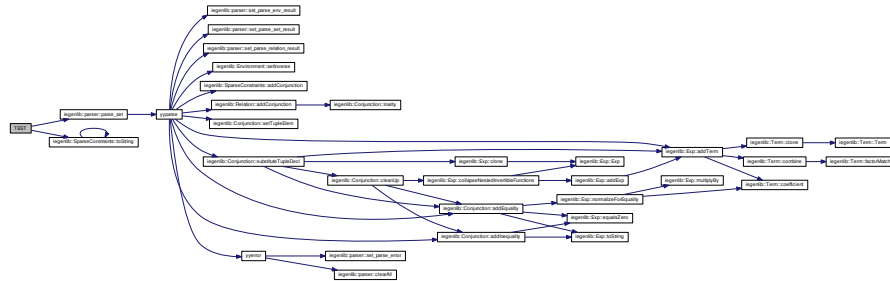
10.7.2.34 TEST ( Parser , SetDuplicateTupleVarConstraintsEXPEQ )

Here is the call graph for this function:



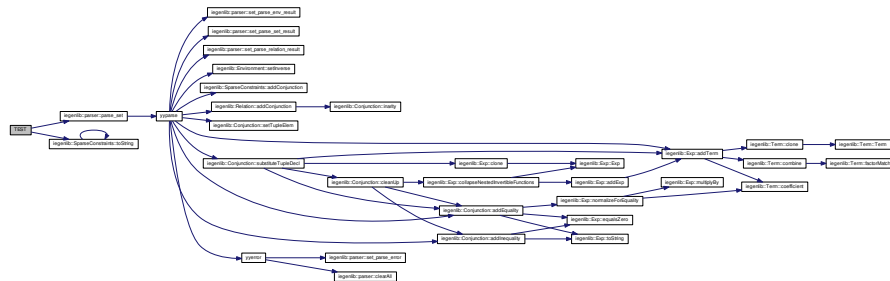
## 10.7.2.35 TEST ( Parser , SetUndeclaredSymbolic )

Here is the call graph for this function:



## 10.7.2.36 TEST ( Parser , SetUndeclaredSymbolicConstraintLAndLT )

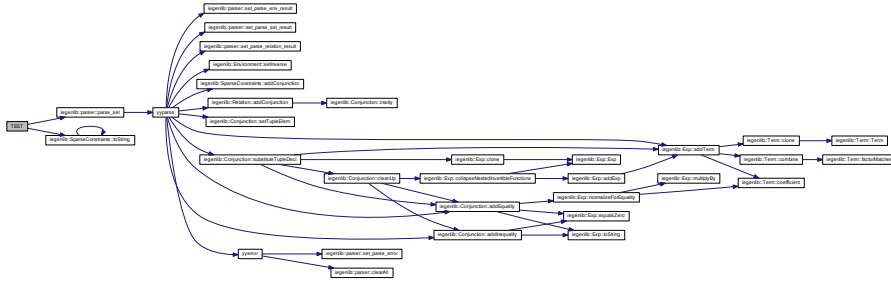
Here is the call graph for this function:





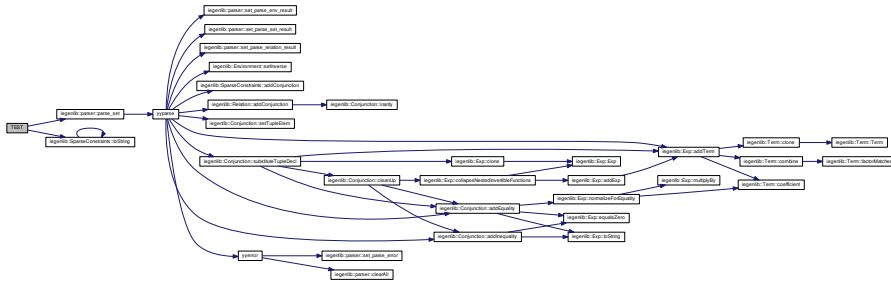
10.7.2.37 TEST ( Parser , SetUndeclaredSymbolicDuplicateTupleVarConstraintLTandLT )

Here is the call graph for this function:



10.7.2.38 TEST ( Parser , SetUndeclaredSymbolicConstraintExpLT )

Here is the call graph for this function:

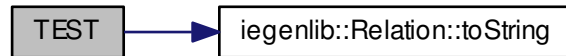






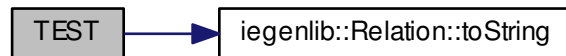
## 10.7.2.43 TEST ( Parser , RelationComposeR1 )

Here is the call graph for this function:



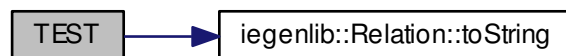
## 10.7.2.44 TEST ( Parser , RelationComposeR2 )

Here is the call graph for this function:



## 10.7.2.45 TEST ( Parser , RelationComposeExpected )

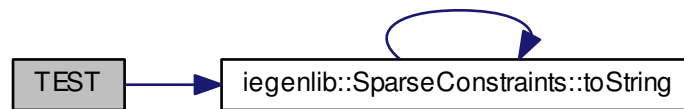
Here is the call graph for this function:





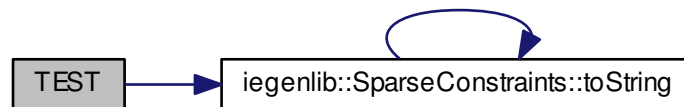
## 10.7.2.48 TEST ( Parser , ISLSetUnion )

Here is the call graph for this function:



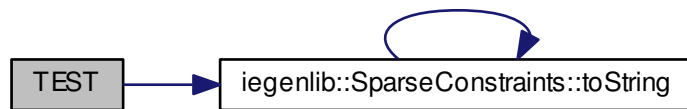
## 10.7.2.49 TEST ( Parser , ISLSetUnion2 )

Here is the call graph for this function:



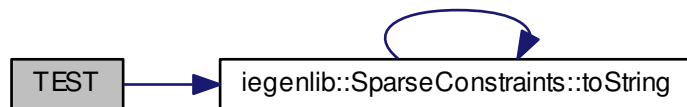
## 10.7.2.50 TEST ( Parser , OmegaSetInSymbolic )

Here is the call graph for this function:



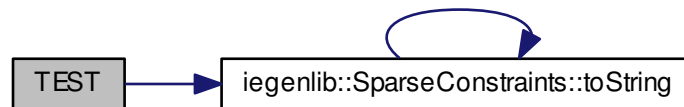
## 10.7.2.51 TEST ( Parser , OmegaSetInSymbolicUnion )

Here is the call graph for this function:



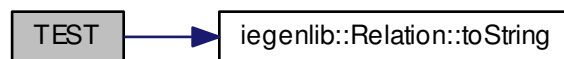
## 10.7.2.52 TEST ( Parser , ISLSetInSymbolicUnion )

Here is the call graph for this function:



## 10.7.2.53 TEST ( Parser , ISLRelationUnion1 )

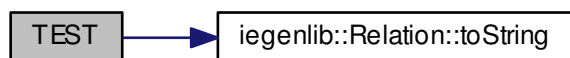
Here is the call graph for this function:





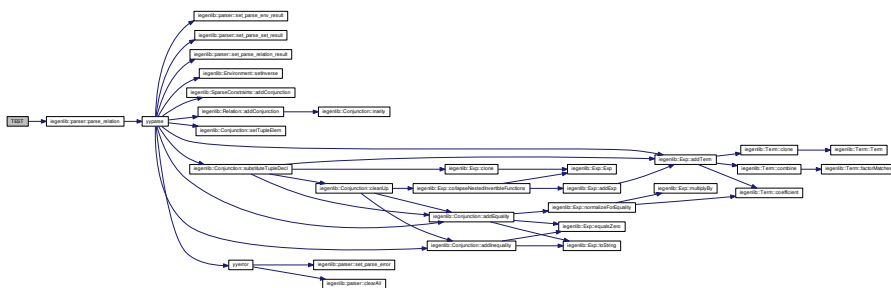
10.7.2.54 TEST ( Parser , ISLRelationInSymbolicUnion1 )

Here is the call graph for this function:



10.7.2.55 TEST ( Parser , ErrorHandlingBug )

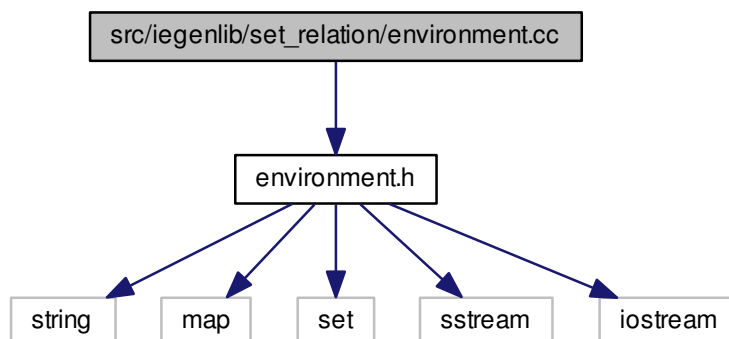
Here is the call graph for this function:



10.8 src/iegenlib/set\_relation/environment.cc File Reference

Implementations of the Environment class.

```
#include "environment.h" Include dependency graph for environment.cc:
```



## Namespaces

- namespace [iegenlib](#)

## Functions

- void [iegenlib::setCurrEnv](#) ()  
*Resets the current environment to empty.*
- void [iegenlib::setCurrEnv](#) (std::string str)  
*Sets the global environment after creating one by parsing the string.*
- void [iegenlib::appendCurrEnv](#) (std::string str)  
*append to this environment*
- std::string [iegenlib::queryInverseCurrEnv](#) (std::string funcName)  
*search this environment for a function inverse*

## Variables

- Environment [iegenlib::currentEnv](#)

### 10.8.1 Detailed Description

Implementations of the Environment class.

Date

Started: 5/1/12 #

Revision:

: last committed revision #

Date:

: date of last committed revision #

Author:

: author of last committed revision

Authors

Michelle Strout and Joe Strout

Copyright (c) 2012, Colorado State University

All rights reserved.

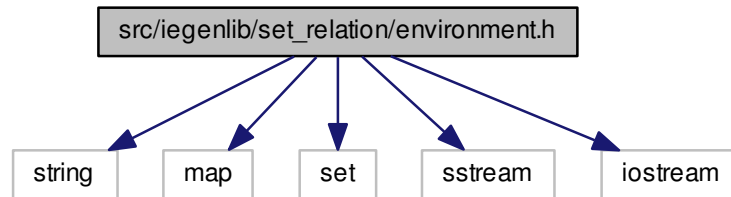
See ../COPYING for details.

## 10.9 src/iegenlib/set\_relation/environment.h File Reference

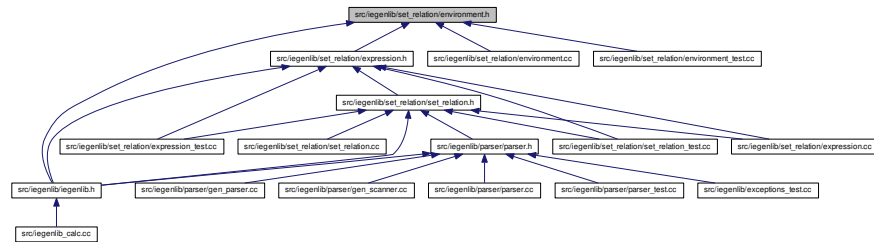
Declarations for the Environment class.

```
#include <string> #include <map> #include <set> #include  
<sstream> #include <iostream> Include dependency graph for environment.h-
```

:



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



## Classes

- class [iegenlib::Environment](#)

## Namespaces

- namespace [iegenlib](#)
- namespace [iegenlib::parser](#)

## Functions

- Environment \* [iegenlib::parser::parse\\_env](#) (std::string env\_string)
- void [iegenlib::setCurrEnv](#) ()

*Resets the current environment to empty.*

- void `iegenlib::setCurrEnv` (std::string str)

*Sets the global environment after creating one by parsing the string.*

- void `iegenlib::appendCurrEnv` (std::string str)

*append to this environment*

- std::string `iegenlib::queryInverseCurrEnv` (std::string funcName)

*search this environment for a function inverse*

### 10.9.1 Detailed Description

Declarations for the Environment class. The Environment class keeps track of information about function inverses, domains, ranges, and symbolic constant constraints.

#### Date

Started: 5/1/12 #

#### Revision:

: last committed revision #

#### Date:

: date of last committed revision #

#### Author:

: author of last committed revision

#### Authors

Michelle Strout

Copyright (c) 2012, Colorado State University

All rights reserved.

See ../../COPYING for details.



**Date**

Started: 5/1/12 #

**Revision:**

: last committed revision #

**Date:**

: date of last committed revision #

**Author:**

: author of last committed revision

**Authors**

Michelle Strout

Copyright (c) 2012, Colorado State University

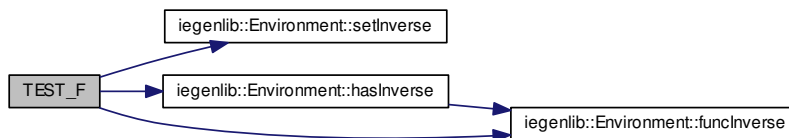
All rights reserved.

See ../../COPYING for details.

## 10.10.2 Function Documentation

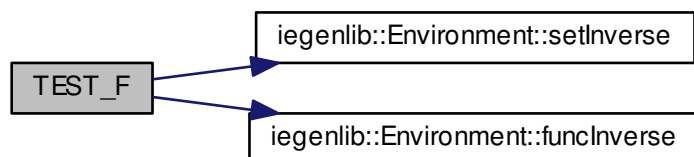
### 10.10.2.1 TEST\_F ( EnvironmentTest , Success )

Here is the call graph for this function:



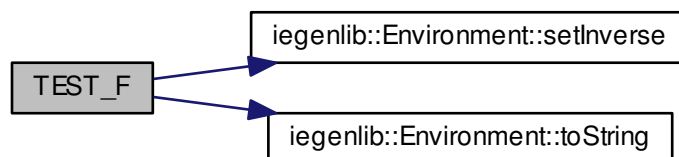
### 10.10.2.2 TEST\_F( EnvironmentTest , NotFound )

Here is the call graph for this function:



### 10.10.2.3 TEST\_F( EnvironmentTest , ToString )

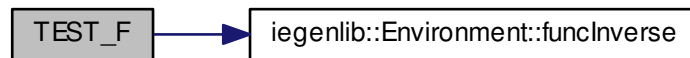
Here is the call graph for this function:





## 10.10.2.4 TEST\_F( EnvironmentTest , EnvParse )

Here is the call graph for this function:



## 10.10.2.5 TEST\_F( EnvironmentTest , EnvAppend )

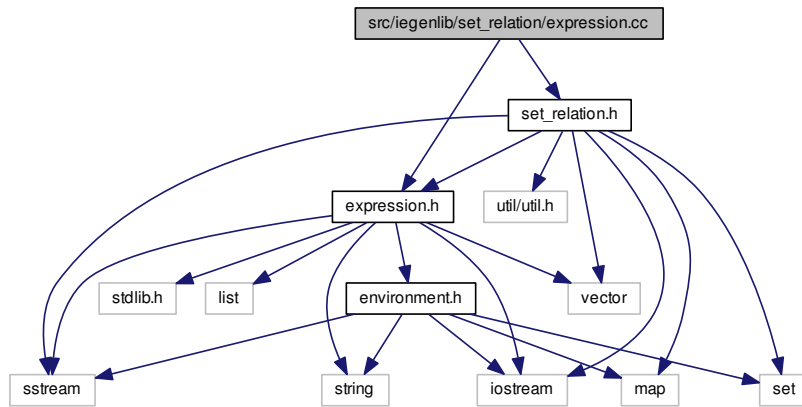
Here is the call graph for this function:



## 10.11 src/iegenlib/set\_relation/expression.cc File Reference

Implementations of the expression classes.

```
#include "expression.h" #include "set_relation.h" Include de-
pendency graph for expression.cc:
```



## Namespaces

- namespace [iegenlib](#)

### 10.11.1 Detailed Description

Implementations of the expression classes.

#### Date

Started: 3/18/2012 #

#### Revision:

490

: last committed revision #

#### Date:

2012-08-09 21:08:51 -0#

: date of last committed revision #

**Author:**

mstrout

: author of last committed revision

**Authors**

Michelle Strout and Joe Strout

Copyright (c) 2012, Colorado State University

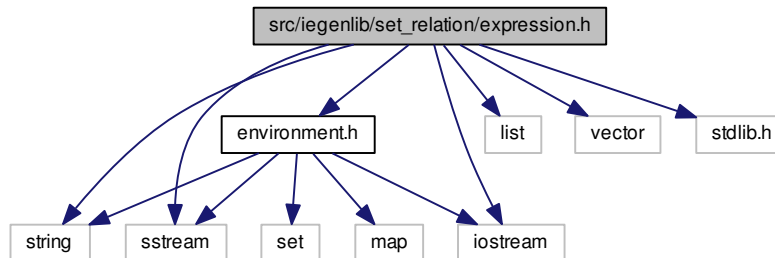
All rights reserved.

See ../../COPYING for details.

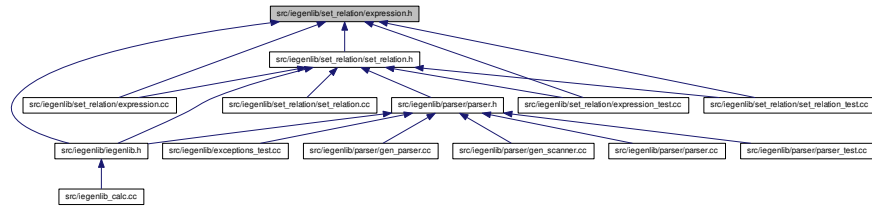
## 10.12 src/iegenlib/set\_relation/expression.h File Reference

Declarations for expression and related classes.

```
#include <string> #include <list> #include <vector> ×  
#include <sstream> #include <stdlib.h> #include <iostream> ×  
#include "environment.h" Include dependency graph for expression.h:
```



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



## Classes

- class [iegenlib::Term](#)  
*A coefficient multiplied by one. Subclasses are multiplied by other entities.*
- class [iegenlib::UFCallTerm](#)  
*Represents a coefficient multiplied by an uninterpreted function call.*
- class [iegenlib::TupleVarTerm](#)  
*Represents a coefficient multiplied by a tuple variable.*
- class [iegenlib::VarTerm](#)  
*Represents a coefficient multiplied by a variable or symbolic constant.*
- class [iegenlib::Exp](#)  
*An affine expression that allows uninterpreted function call terms.*

## Namespaces

- namespace [iegenlib](#)

### 10.12.1 Detailed Description

Declarations for expression and related classes. The expression classes represent affine expressions with uninterpreted function calls as terms allowed.

#### Date

Started: 3/18/2012 #

#### Revision:

487

: last committed revision #

Date:

2012-08-09 13:50:27 -0#

: date of last committed revision #

Author:

mstrout

: author of last committed revision

Authors

Michelle Strout and Joe Strout

Copyright (c) 2012, Colorado State University

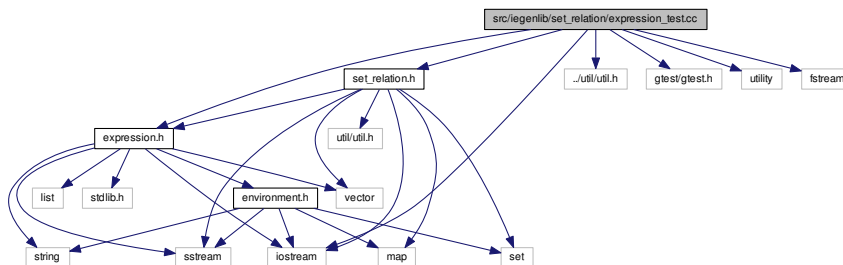
All rights reserved.

See ../../COPYING for details.

## 10.13 src/iegenlib/set\_relation/expression\_test.cc File Reference

Test for the Expression classes and Term classes.

```
#include "expression.h" #include "set_relation.h" #include
"../util/util.h" #include <gtest/gtest.h> #include <utility> ×
#include <iostream> #include <fstream> Include dependency graph
for expression_test.cc:
```



## Classes

- class [ExpTest](#)

## Functions

- [TEST\\_F \(ExpTest, TermMethods\)](#)
- [TEST\\_F \(ExpTest, VarTermMethods\)](#)
- [TEST\\_F \(ExpTest, TupleVarTermMethods\)](#)
- [TEST\\_F \(ExpTest, UFCallTermMethods\)](#)
- [TEST\\_F \(ExpTest, ExpMethods\)](#)
- [TEST\\_F \(ExpTest, ExposeUFCallNotElimBug\)](#)
- [TEST\\_F \(ExpTest, ExpMethodsAdvanced\)](#)
- [TEST\\_F \(ExpTest, ExpSolving\)](#)
- [TEST\\_F \(ExpTest, ExpDependsOn\)](#)
- [TEST\\_F \(ExpTest, ExpTermCompare\)](#)
- [TEST\\_F \(ExpTest, JIRA\\_85\)](#)
- [TEST\\_F \(ExpTest, ExpRemapLocs\)](#)
- [TEST\\_F \(ExpTest, ExplsConst\)](#)
- [TEST\\_F \(ExpTest, PrettyPrintString\)](#)
- [TEST\\_F \(ExpTest, ToDotString\)](#)

### 10.13.1 Detailed Description

Test for the Expression classes and Term classes. This file is to test all of the Expression and Term classes, along with the related classes: EqExp, IneqExp, UFCallTerm, VarTerm, and TupleVarTerm.

#### Date

Started: 3/18/2012 #

#### Revision:

493

: last committed revision #

#### Date:

2012-08-09 21:16:33 -0#

: date of last committed revision #

**Author:**

mstrout

: author of last committed revision

**Authors**

Michelle Strout and Joe Strout

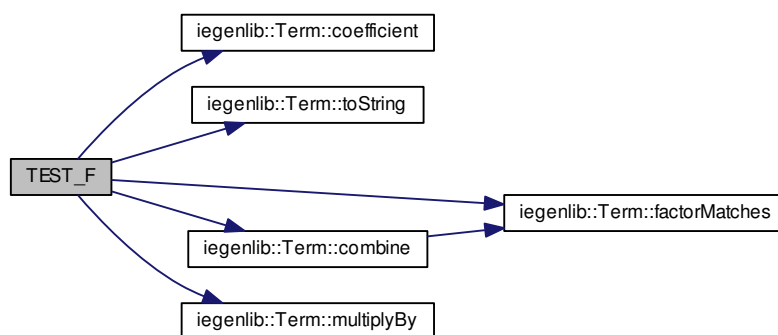
Copyright (c) 2012, Colorado State University

All rights reserved.

See ../../COPYING for details.

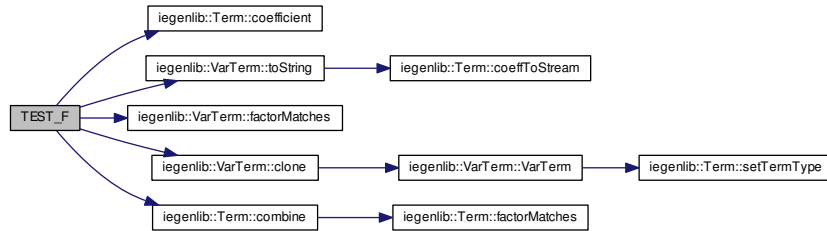
**10.13.2 Function Documentation****10.13.2.1 TEST\_F ( ExpTest , TermMethods )**

Here is the call graph for this function:



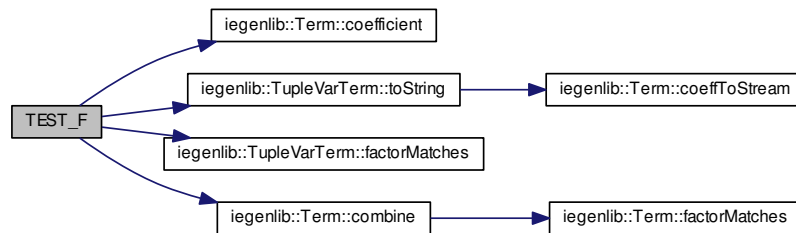
### 10.13.2.2 TEST\_F( ExpTest , VarTermMethods )

Here is the call graph for this function:



### 10.13.2.3 TEST\_F( ExpTest , TupleVarTermMethods )

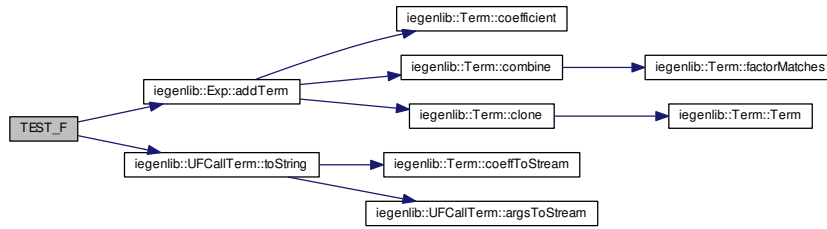
Here is the call graph for this function:





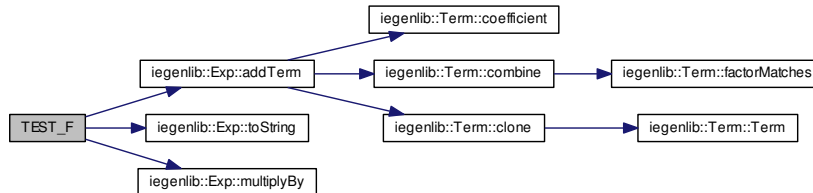
## 10.13.2.4 TEST\_F( ExpTest , UFCallTermMethods )

Here is the call graph for this function:



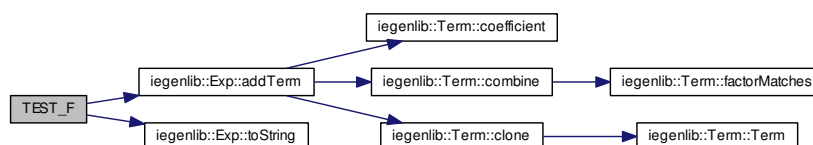
## 10.13.2.5 TEST\_F( ExpTest , ExpMethods )

Here is the call graph for this function:



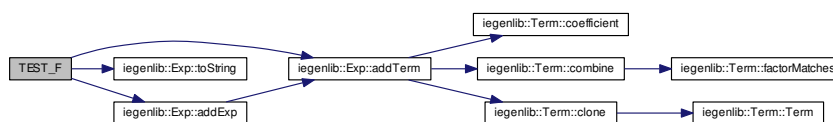
## 10.13.2.6 TEST\_F( ExpTest , ExposeUFCallNotElimBug )

Here is the call graph for this function:



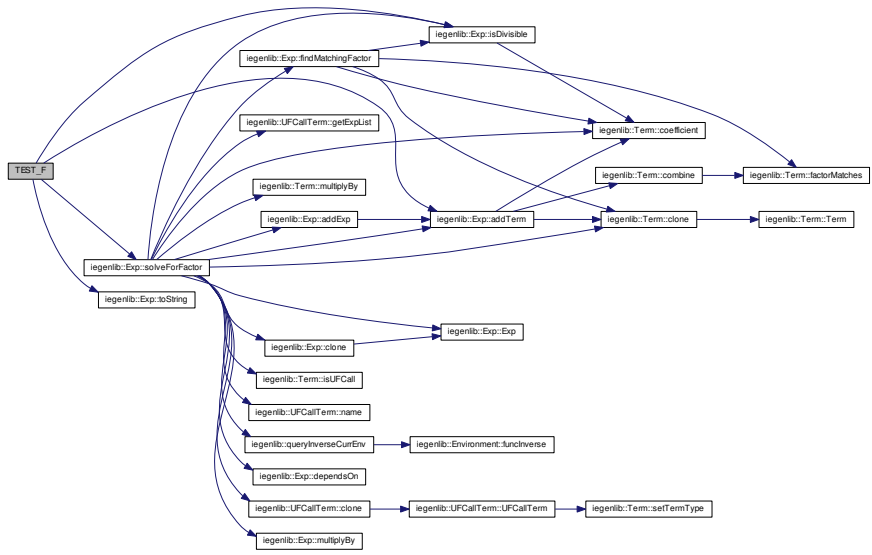
## 10.13.2.7 TEST\_F( ExpTest , ExpMethodsAdvanced )

Here is the call graph for this function:



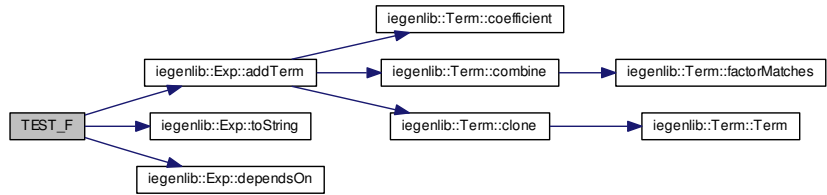
10.13.2.8 TEST\_F( ExpTest , ExpSolving )

Here is the call graph for this function:



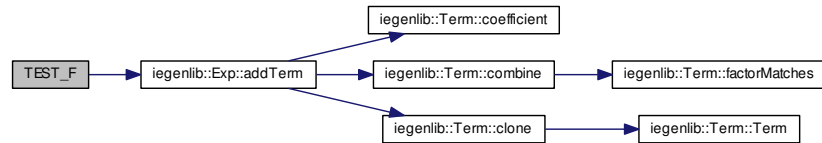
10.13.2.9 TEST\_F( ExpTest , ExpDependsOn )

Here is the call graph for this function:



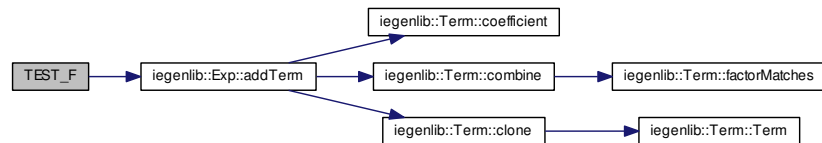
## 10.13.2.10 TEST\_F( ExpTest , ExpTermCompare )

Here is the call graph for this function:



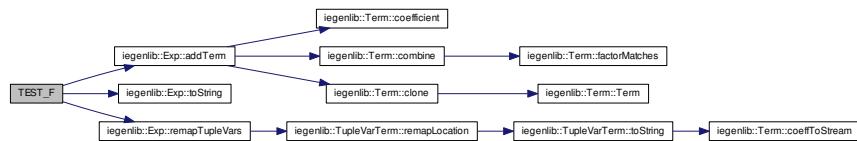
## 10.13.2.11 TEST\_F( ExpTest , JIRA.85 )

Here is the call graph for this function:



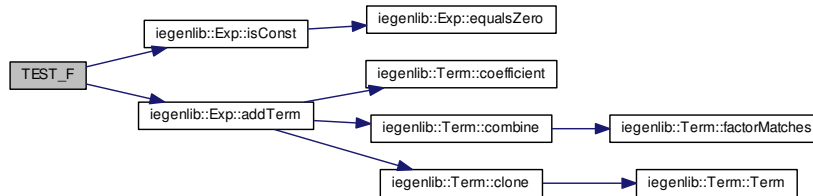
## 10.13.2.12 TEST\_F( ExpTest , ExpRemapLocs )

Here is the call graph for this function:



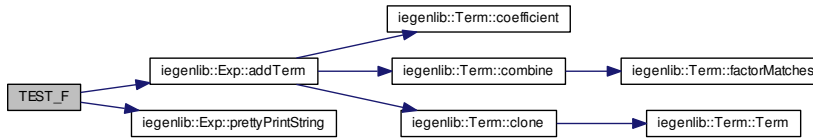
## 10.13.2.13 TEST\_F ( ExpTest , ExplsConst )

Here is the call graph for this function:



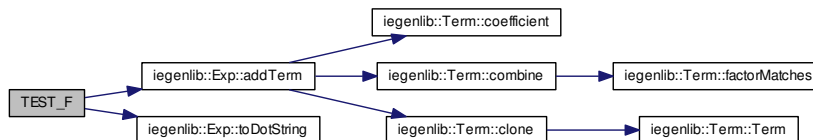
## 10.13.2.14 TEST\_F ( ExpTest , PrettyPrintString )

Here is the call graph for this function:



## 10.13.2.15 TEST\_F ( ExpTest , ToDotString )

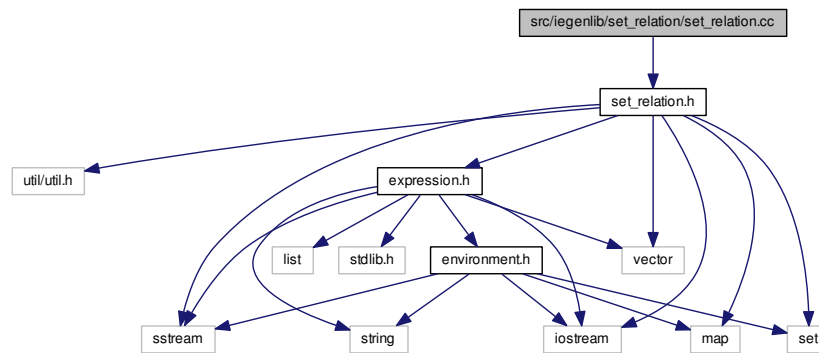
Here is the call graph for this function:



## 10.14 src/iegenlib/set\_relation/set\_relation.cc File Reference

Implementations of the Set and Relation classes.

`#include "set_relation.h"` Include dependency graph for `set_relation.cc`:



### Namespaces

- namespace [iegenlib](#)

#### 10.14.1 Detailed Description

Implementations of the Set and Relation classes.

#### Date

Started: 3/28/12 #

#### Revision:

496

: last committed revision #

#### Date:

2012-08-22 22:30:57 -0#

: date of last committed revision #

Author:

mstrout

: author of last committed revision

Authors

Michelle Strout and Joe Strout

Copyright (c) 2012, Colorado State University

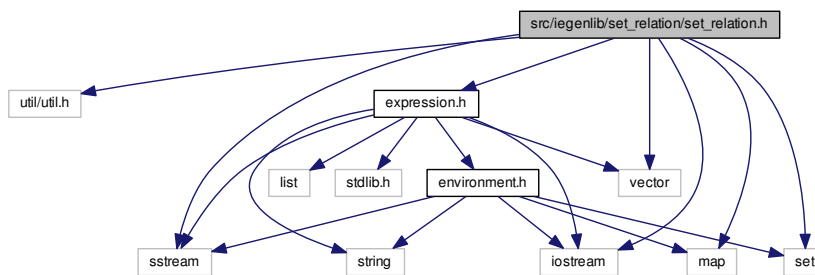
All rights reserved.

See ../COPYING for details.

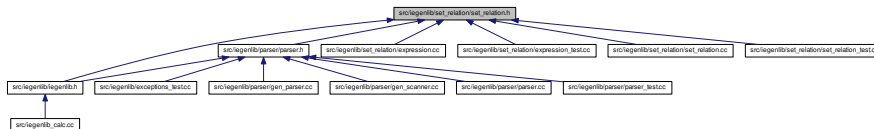
10.15 src/iegenlib/set\_relation/set\_relation.h File Reference

Declarations for the set and relation classes and classes they contain except for the Exp and Term classes.

```
#include <util/util.h> #include "expression.h" #include
<set> #include <vector> #include <map> #include <sstream> ×
#include <iostream> Include dependency graph for set_relation.h:
```



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



## Classes

- class [iegenlib::TupleElemDecl](#)  
*Info about a tuple element, which is a constant or a variable.*
- class [iegenlib::Conjunction](#)  
*Class containing sets of all the equalities and inequalities.*
- class [iegenlib::SparseConstraints](#)  
*Base class that contains the conjunctions and a pointer to an environment.*
- class [iegenlib::Set](#)  
*A [SparseConstraints](#) class that represents a [Set](#).*
- class [iegenlib::Relation](#)  
*A [SparseConstraints](#) class that represents a [Relation](#).*

## Namespaces

- namespace [iegenlib](#)
- namespace [iegenlib::parser](#)

## Functions

- Set \* [iegenlib::parser::parse\\_set](#) (std::string set\_string)
- Relation \* [iegenlib::parser::parse\\_relation](#) (std::string relation\_string)

### 10.15.1 Detailed Description

Declarations for the set and relation classes and classes they contain except for the Exp and Term classes. The Set and Relation classes represent integer tuple sets and relations with sets of conjunctions that contain sets of affine inequality and equality constraints. The constraints can include uninterpreted function symbol terms.

#### Date

Started: 3/28/12 #

#### Revision:

490

: last committed revision #

#### Date:

2012-08-09 21:08:51 -0#



: date of last committed revision #

**Author:**

mstrout

: author of last committed revision

**Authors**

Michelle Strout

Copyright (c) 2012, Colorado State University

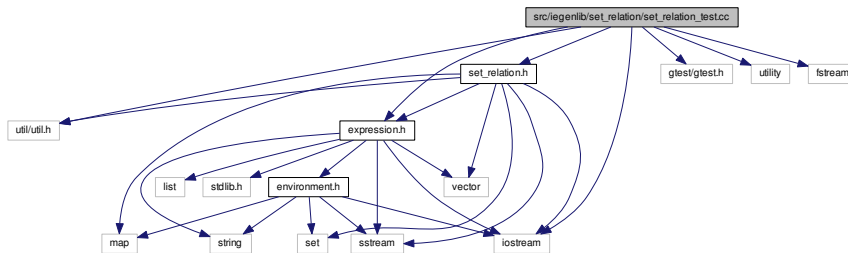
All rights reserved.

See ../COPYING for details.

## 10.16 src/iegenlib/set\_relation/set\_relation\_test.cc File Reference

Set and Relation tests.

```
#include "set_relation.h" #include "expression.h" #include
<util/util.h> #include <gtest/gtest.h> #include <utility> ×
#include <fstream> #include <iostream> Include dependency graph
for set_relation_test.cc:
```



### Classes

- class [SetRelationTest](#)

### Functions

- [TEST\\_F](#) ([SetRelationTest](#), [SetUpTest](#))

- [TEST\\_F \(SetRelationTest, ConjunctionConstruction\)](#)
- [TEST\\_F \(SetRelationTest, ConjunctionToDot\)](#)
- [TEST\\_F \(SetRelationTest, SparseConstraints\)](#)
- [TEST\\_F \(SetRelationTest, Set\)](#)
- [TEST\\_F \(SetRelationTest, SetPrettyPrintString\)](#)
- [TEST\\_F \(SetRelationTest, Relation\)](#)
- [TEST\\_F \(SetRelationTest, RelationPrettyPrintString\)](#)
- [TEST\\_F \(SetRelationTest, RelationWithUFCall\)](#)
- [TEST\\_F \(SetRelationTest, SolveForFactor\)](#)
- [TEST\\_F \(SetRelationTest, FindFunction\)](#)
- [TEST\\_F \(SetRelationTest, ExpressionSignBug\)](#)
- [TEST\\_F \(SetRelationTest, EmptyExpressionBug\)](#)
- [TEST\\_F \(SetRelationTest, SRToDot\)](#)
- [TEST\\_F \(SetRelationTest, CompositionIdentity\)](#)
- [TEST\\_F \(SetRelationTest, CompositionBreakIdentityOnly\)](#)
- [TEST\\_F \(SetRelationTest, CompositionInverses\)](#)
- [TEST\\_F \(SetRelationTest, CompositionVariousArities\)](#)
- [TEST\\_F \(SetRelationTest, ComposeNeedsRenaming\)](#)
- [TEST\\_F \(SetRelationTest, ComposeWithUFC\)](#)
- [TEST\\_F \(SetRelationTest, ComposeWithUFCNeedsInverse\)](#)
- [TEST\\_F \(SetRelationTest, CompositionCombo\)](#)
- [TEST\\_F \(SetRelationTest, CompositionBothFunctionsWithConstraints\)](#)
- [TEST\\_F \(SetRelationTest, ParseTest1\)](#)
- [TEST\\_F \(SetRelationTest, ComposeWithConstants\)](#)
- [TEST\\_F \(SetRelationTest, CheckDupTupleVars\)](#)
- [TEST\\_F \(SetRelationTest, FindConstantFunctions\)](#)
- [TEST\\_F \(SetRelationTest, ApplySpMV\)](#)
- [TEST\\_F \(SetRelationTest, Union\)](#)
- [TEST\\_F \(SetRelationTest, Inverse\)](#)
- [TEST\\_F \(SetRelationTest, ComposeWithInnerConstants\)](#)
- [TEST\\_F \(SetRelationTest, ComposeWithUnions\)](#)
- [TEST\\_F \(SetRelationTest, ComposeWithOtherConstraints\)](#)
- [TEST\\_F \(SetRelationTest, CompositionWithNestedFunctions\)](#)
- [TEST\\_F \(SetRelationTest, ComposeWithUFCallConstraints\)](#)
- [TEST\\_F \(SetRelationTest, RemoveDuplicateConstraints\)](#)
- [TEST\\_F \(SetRelationTest, RemoveManyDups\)](#)
- [TEST\\_F \(SetRelationTest, RemoveColon\)](#)
- [TEST\\_F \(SetRelationTest, UsingEnvironmentParsing\)](#)
- [TEST\\_F \(SetRelationTest, LCPC12SubmissionComposeNotFunc\)](#)
- [TEST\\_F \(SetRelationTest, OrderingConstraints\)](#)
- [TEST\\_F \(SetRelationTest, IEGRONE64Apply\)](#)
- [TEST\\_F \(SetRelationTest, IEGRONE41Zeros\)](#)
- [TEST\\_F \(SetRelationTest, IEGRONE72CollapsingInverseFuncs\)](#)

- [TEST\\_F \(SetRelationTest, MoldynFSTExample\)](#)
- [TEST\\_F \(SetRelationTest, MoldynFSTExampleInverse\)](#)
- [TEST\\_F \(SetRelationTest, AnandComposeExample\)](#)
- [TEST\\_F \(SetRelationTest, MoldynManyTests\)](#)

### 10.16.1 Detailed Description

Set and Relation tests. This file is to test all of the Set and Relation classes, along with the related Conjunction class.

#### Date

Started: 3/28/12 #

#### Revision:

512

: last committed revision #

#### Date:

2012-09-19 14:48:35 -0#

: date of last committed revision #

#### Author:

mstrout

: author of last committed revision

#### Authors

Michelle Strout and Joseph Strout

Copyright (c) 2012, Colorado State University

All rights reserved.

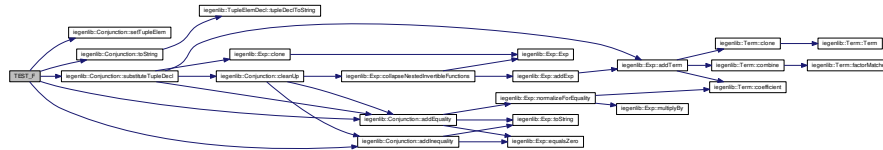
See ../COPYING for details.

### 10.16.2 Function Documentation

#### 10.16.2.1 TEST\_F ( SetRelationTest , SetUpTest )

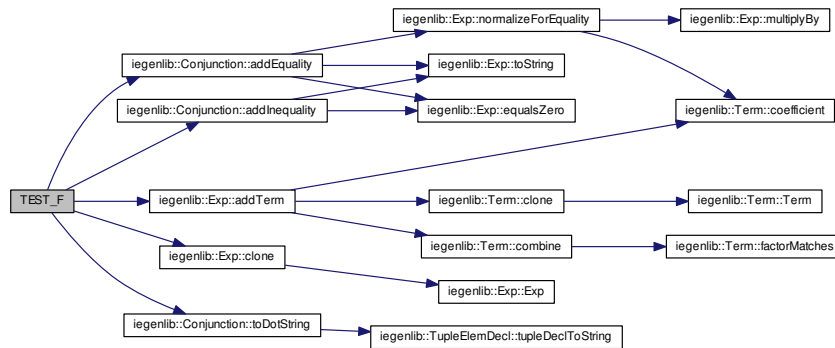
### 10.16.2.2 TEST\_F ( SetRelationTest , ConjunctionConstruction )

Here is the call graph for this function:



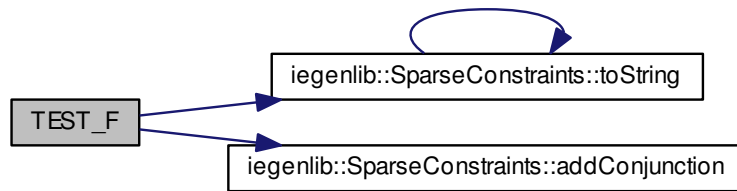
### 10.16.2.3 TEST\_F ( SetRelationTest , ConjunctionToDot )

Here is the call graph for this function:



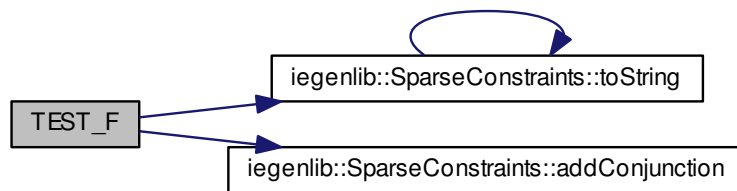
## 10.16.2.4 TEST\_F( SetRelationTest , SparseConstraints )

Here is the call graph for this function:



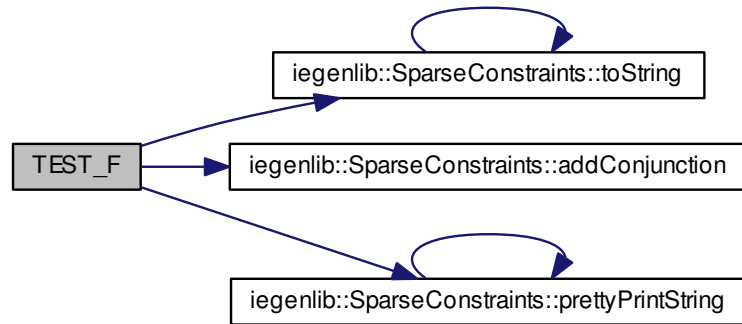
## 10.16.2.5 TEST\_F( SetRelationTest , Set )

Here is the call graph for this function:



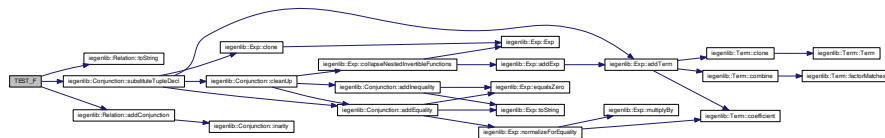
## 10.16.2.6 TEST\_F ( SetRelationTest , SetPrettyPrintString )

Here is the call graph for this function:



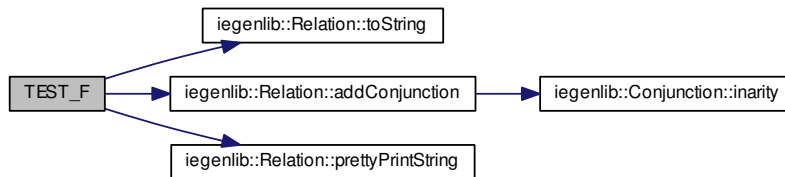
## 10.16.2.7 TEST\_F ( SetRelationTest , Relation )

Here is the call graph for this function:



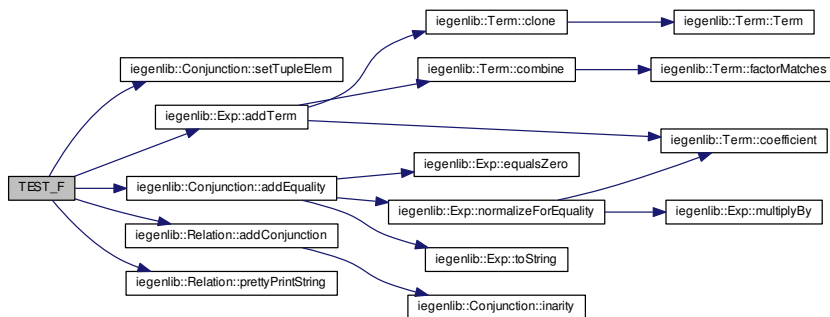
## 10.16.2.8 TEST\_F ( SetRelationTest , RelationPrettyPrintString )

Here is the call graph for this function:



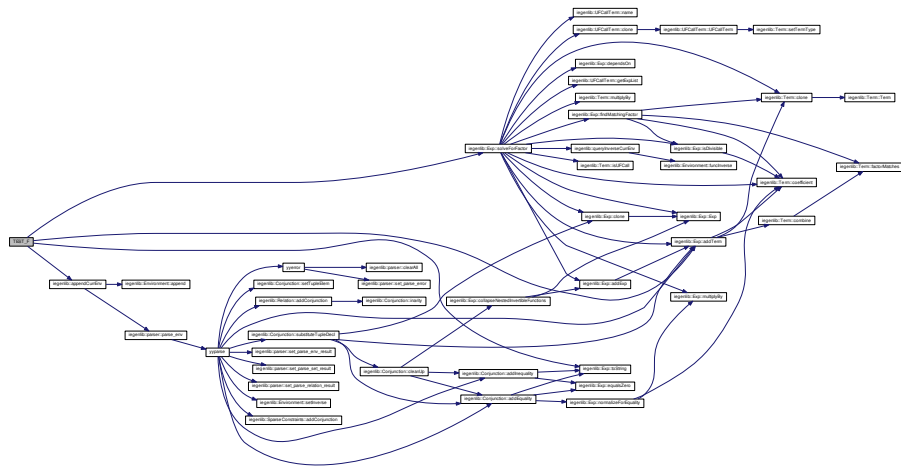
## 10.16.2.9 TEST\_F ( SetRelationTest , RelationWithUFCall )

Here is the call graph for this function:



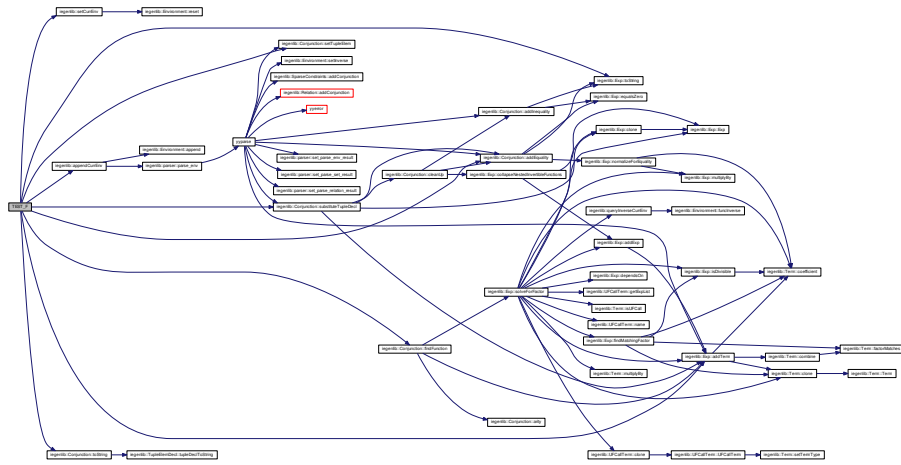
10.16.2.10 TEST\_F( SetRelationTest , SolveForFactor )

Here is the call graph for this function:



10.16.2.11 TEST\_F( SetRelationTest , FindFunction )

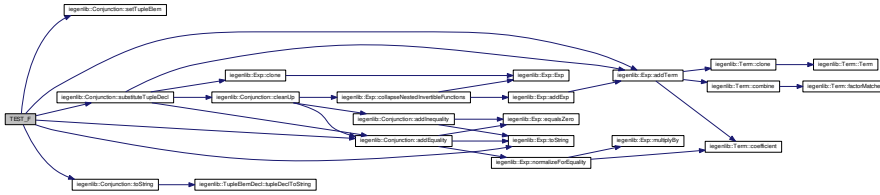
Here is the call graph for this function:





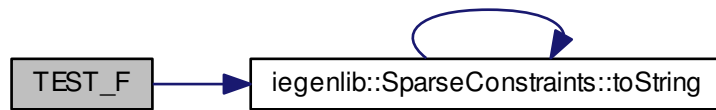
## 10.16.2.12 TEST\_F ( SetRelationTest , ExpressionSignBug )

Here is the call graph for this function:



## 10.16.2.13 TEST\_F ( SetRelationTest , EmptyExpressionBug )

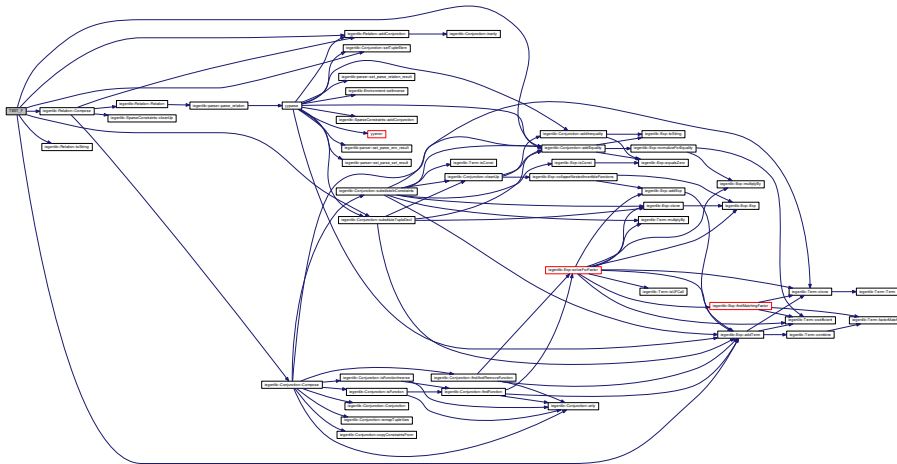
Here is the call graph for this function:





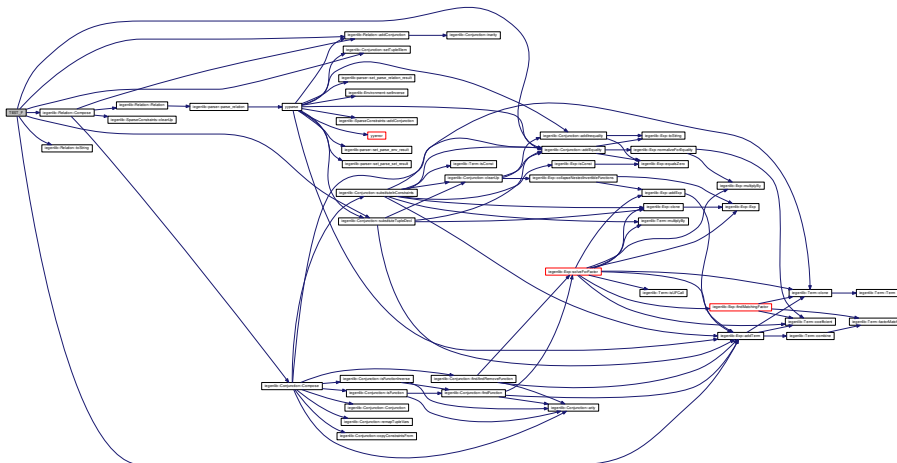
10.16.2.16 TEST\_F ( SetRelationTest , CompositionBreakIdentityOnly )

Here is the call graph for this function:



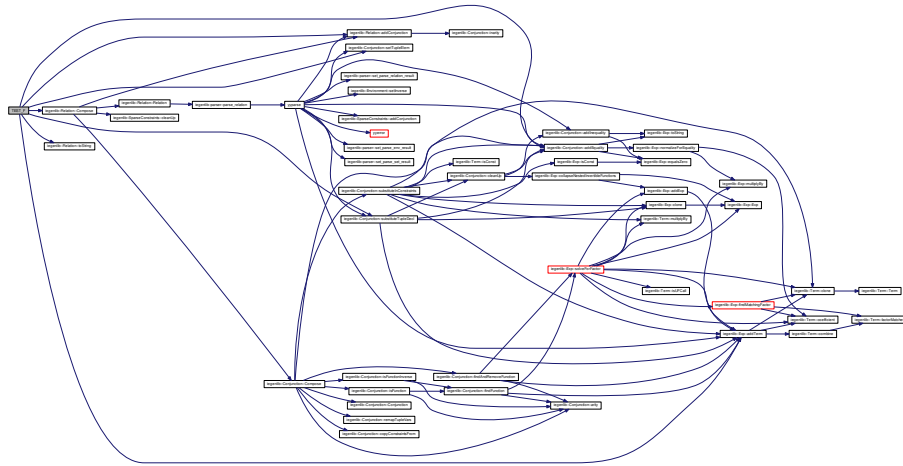
10.16.2.17 TEST\_F ( SetRelationTest , CompositionInverses )

Here is the call graph for this function:



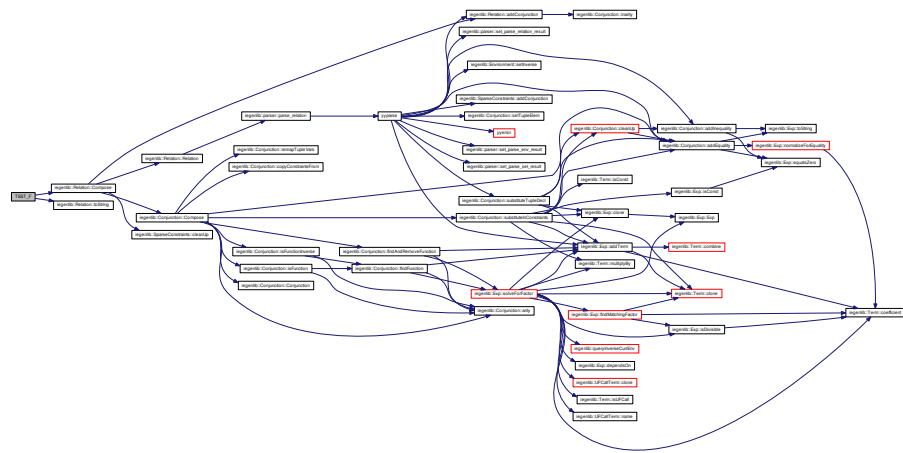
10.16.2.18 TEST\_F( SetRelationTest , CompositionVariousArities )

Here is the call graph for this function:



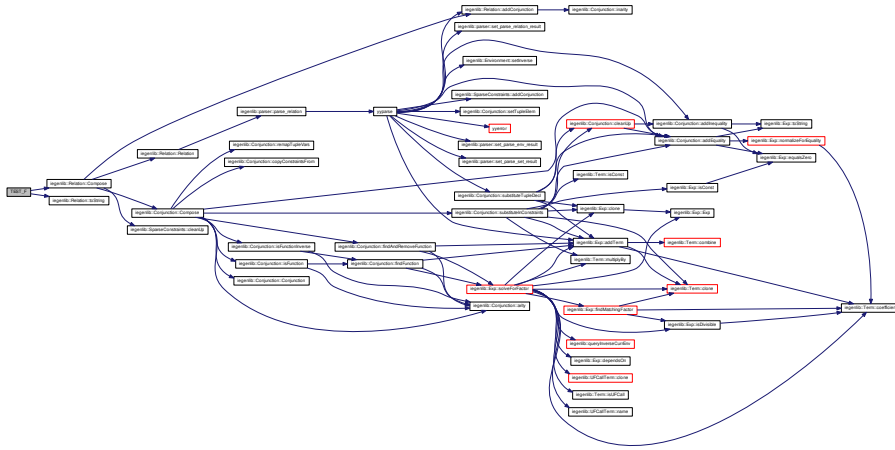
10.16.2.19 TEST\_F( SetRelationTest , ComposeNeedsRenaming )

Here is the call graph for this function:



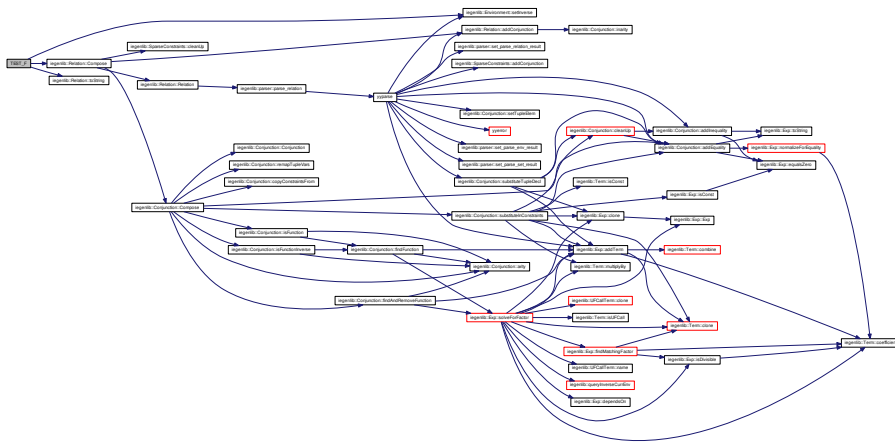
10.16.2.20 TEST\_F ( SetRelationTest , ComposeWithUFC )

Here is the call graph for this function:



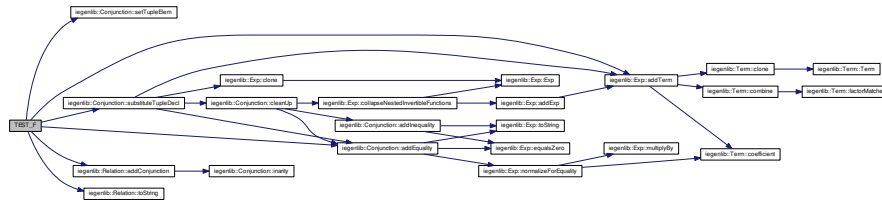
10.16.2.21 TEST\_F ( SetRelationTest , ComposeWithUFCNeedsInverse )

Here is the call graph for this function:



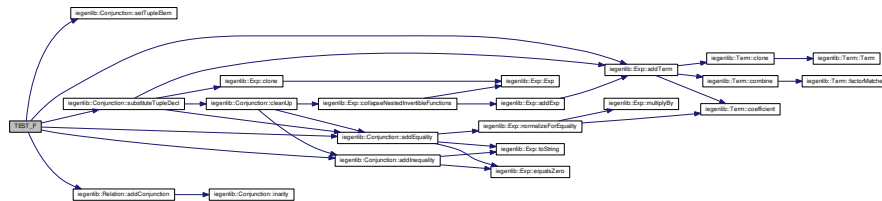
## 10.16.2.22 TEST\_F ( SetRelationTest , CompositionCombo )

Here is the call graph for this function:



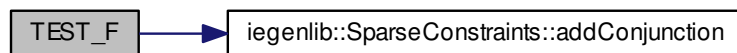
## 10.16.2.23 TEST\_F ( SetRelationTest , CompositionBothFunctionsWithConstraints )

Here is the call graph for this function:



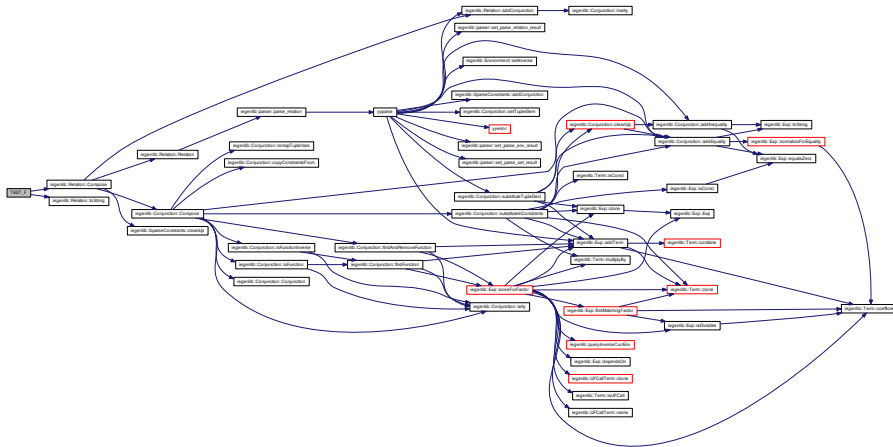
## 10.16.2.24 TEST\_F ( SetRelationTest , ParseTest1 )

Here is the call graph for this function:



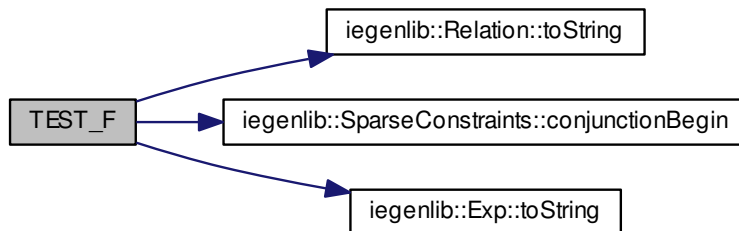
10.16.2.25 TEST\_F ( SetRelationTest , ComposeWithConstants )

Here is the call graph for this function:



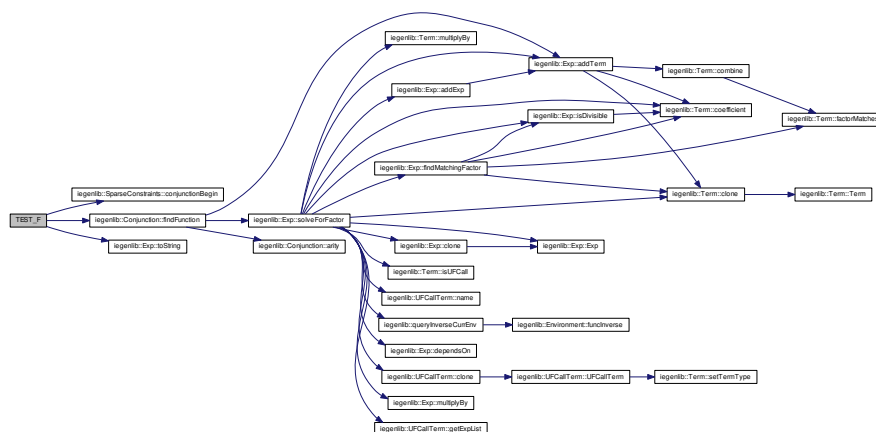
10.16.2.26 TEST\_F ( SetRelationTest , CheckDupTupleVars )

Here is the call graph for this function:



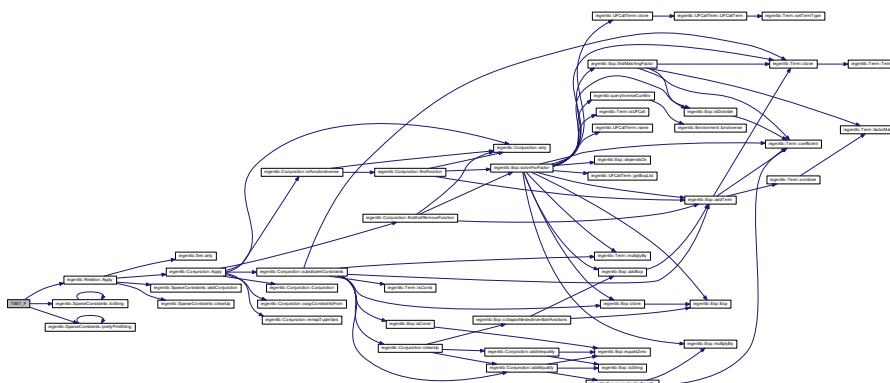
### 10.16.2.27 TEST\_F ( SetRelationTest , FindConstantFunctions )

Here is the call graph for this function:



### 10.16.2.28 TEST\_F ( SetRelationTest , ApplySpMV )

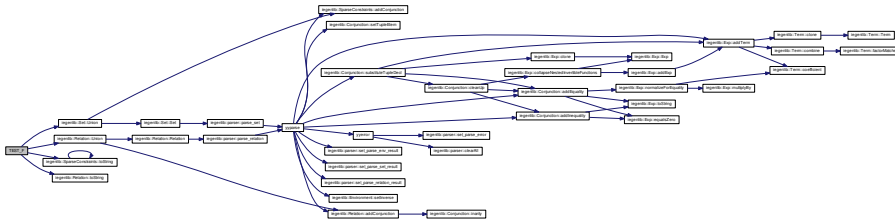
Here is the call graph for this function:





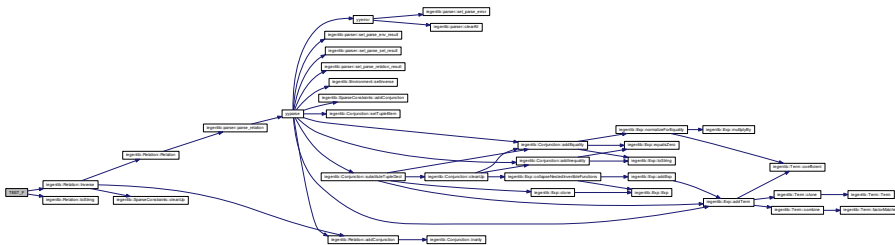
10.16.2.29 TEST\_F ( SetRelationTest , Union )

Here is the call graph for this function:



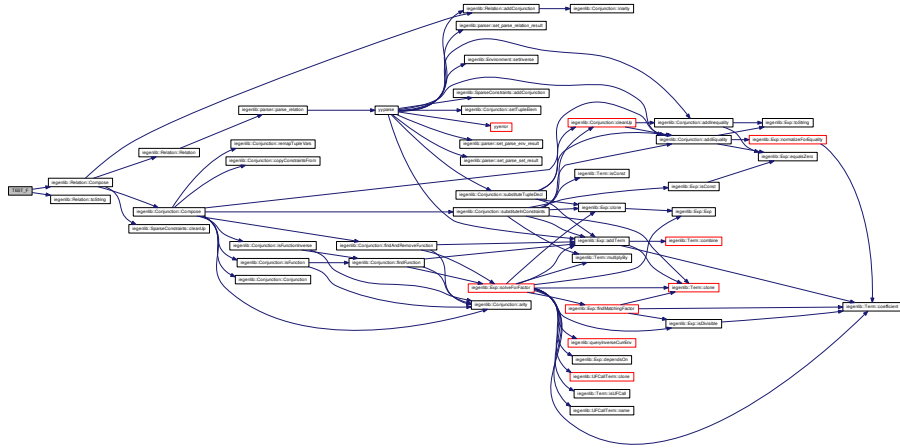
10.16.2.30 TEST\_F ( SetRelationTest , Inverse )

Here is the call graph for this function:



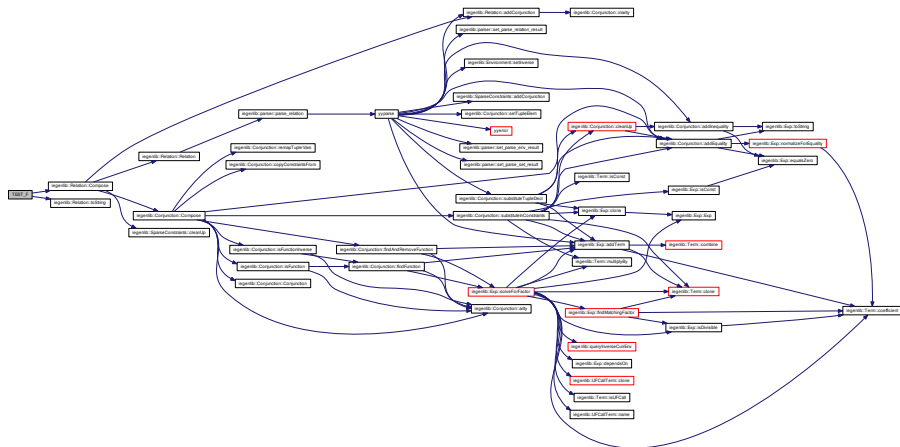
### 10.16.2.31 TEST\_F( SetRelationTest , ComposeWithInnerConstants )

Here is the call graph for this function:



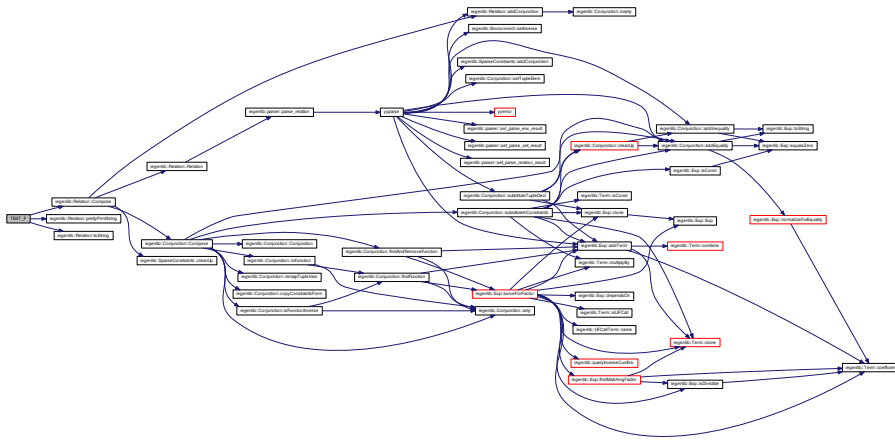
### 10.16.2.32 TEST\_F( SetRelationTest , ComposeWithUnions )

Here is the call graph for this function:



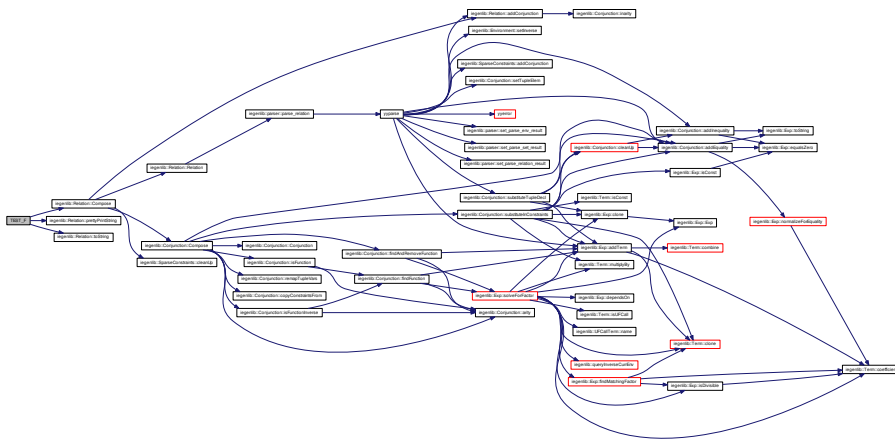
10.16.2.33 TEST\_F ( SetRelationTest , ComposeWithOtherConstraints )

Here is the call graph for this function:



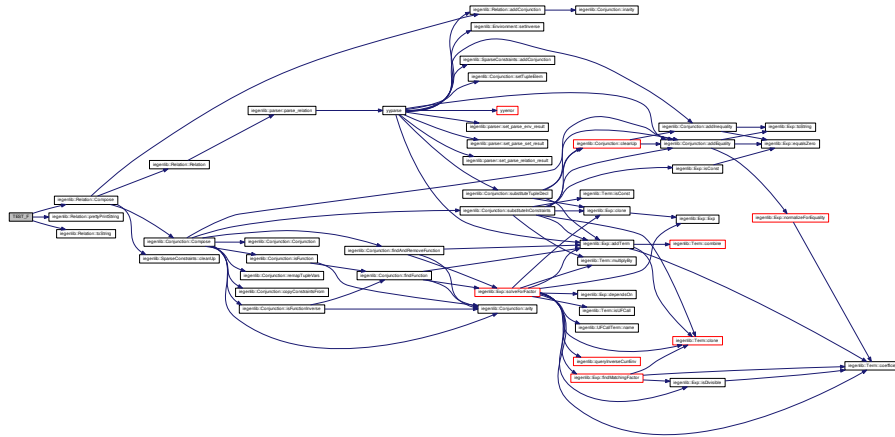
10.16.2.34 TEST\_F ( SetRelationTest , CompositionWithNestedFunctions )

Here is the call graph for this function:



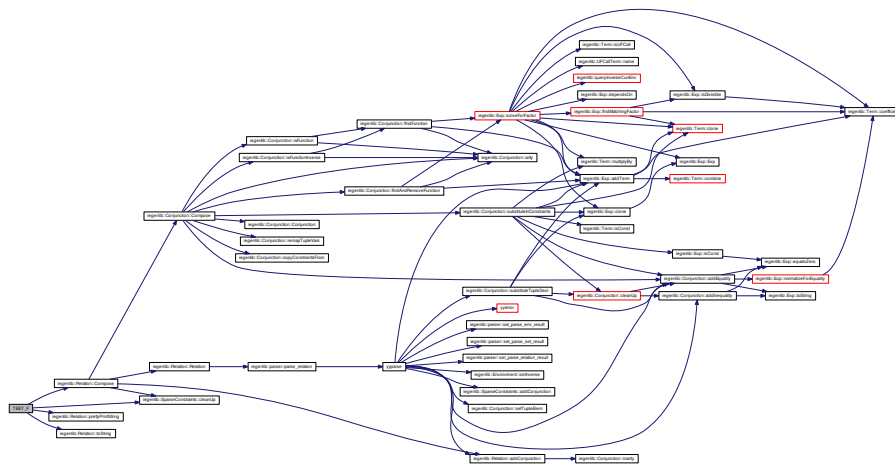
10.16.2.35 TEST\_F( SetRelationTest , ComposeWithUFCallConstraints )

Here is the call graph for this function:



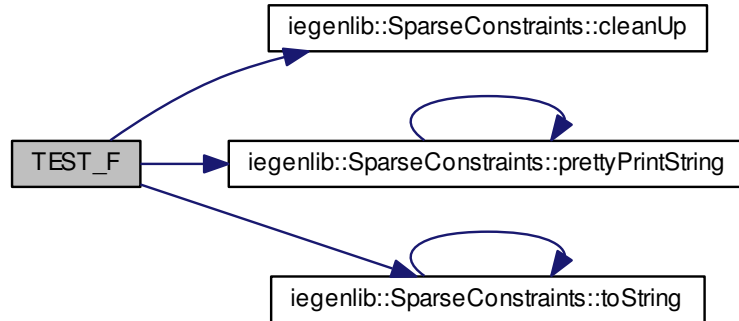
10.16.2.36 TEST\_F( SetRelationTest , RemoveDuplicateConstraints )

Here is the call graph for this function:



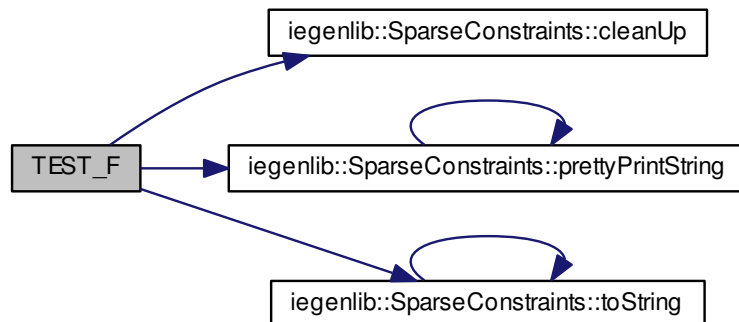
## 10.16.2.37 TEST\_F ( SetRelationTest , RemoveManyDups )

Here is the call graph for this function:



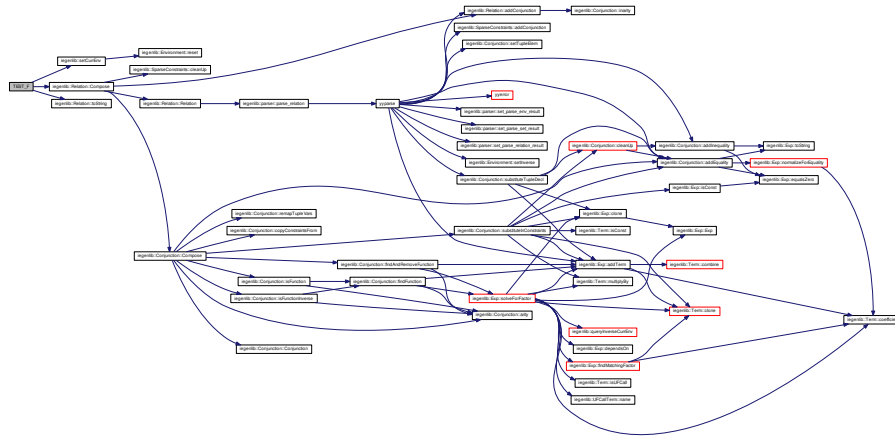
## 10.16.2.38 TEST\_F ( SetRelationTest , RemoveColon )

Here is the call graph for this function:



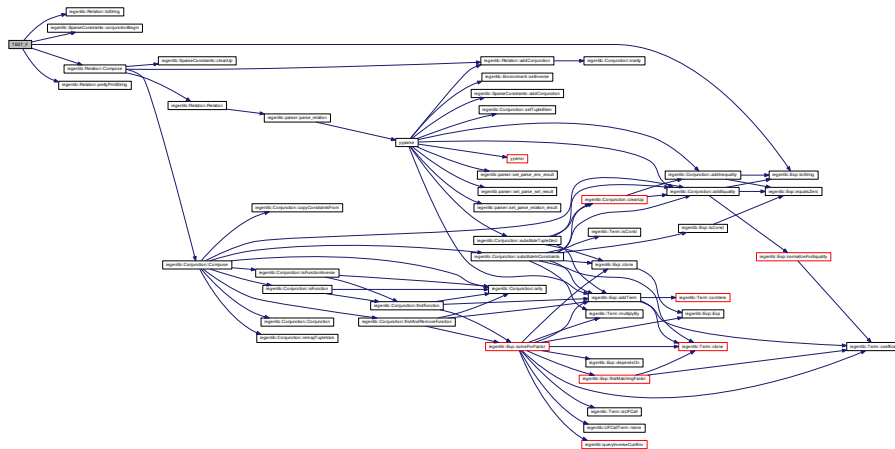
10.16.2.39 TEST\_F( SetRelationTest , UsingEnvironmentParsing )

Here is the call graph for this function:



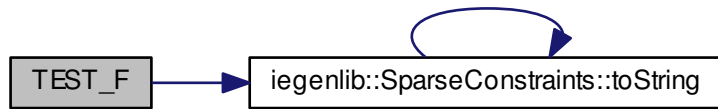
10.16.2.40 TEST\_F( SetRelationTest , LCPC12SubmissionComposeNotFunc )

Here is the call graph for this function:



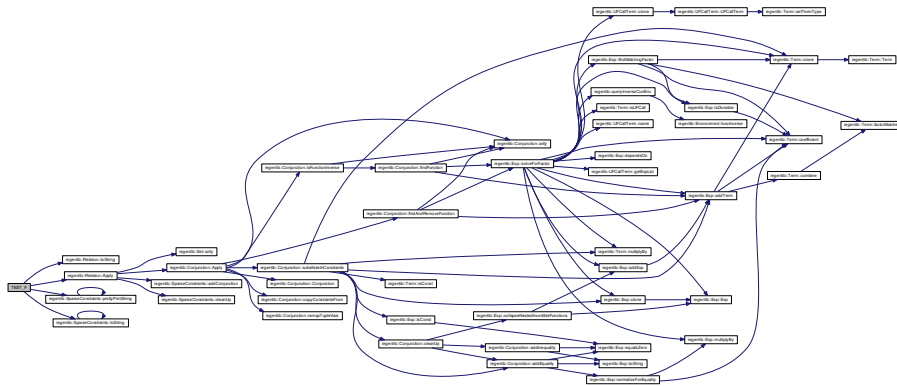
10.16.2.41 TEST\_F ( SetRelationTest , OrderingConstraints )

Here is the call graph for this function:



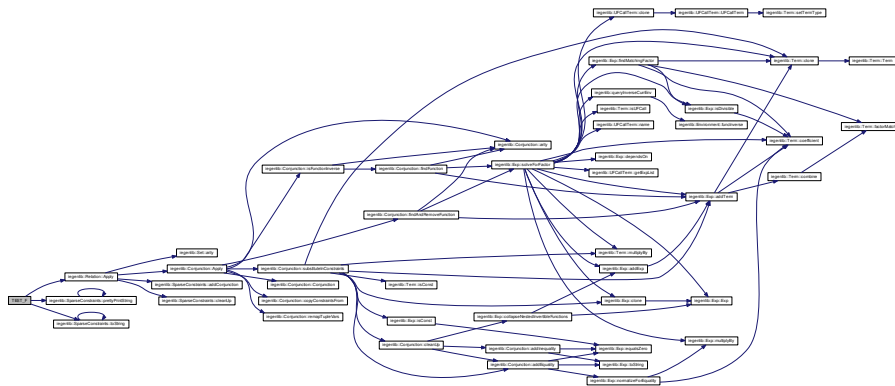
10.16.2.42 TEST\_F ( SetRelationTest , IEGRONE64Apply )

Here is the call graph for this function:



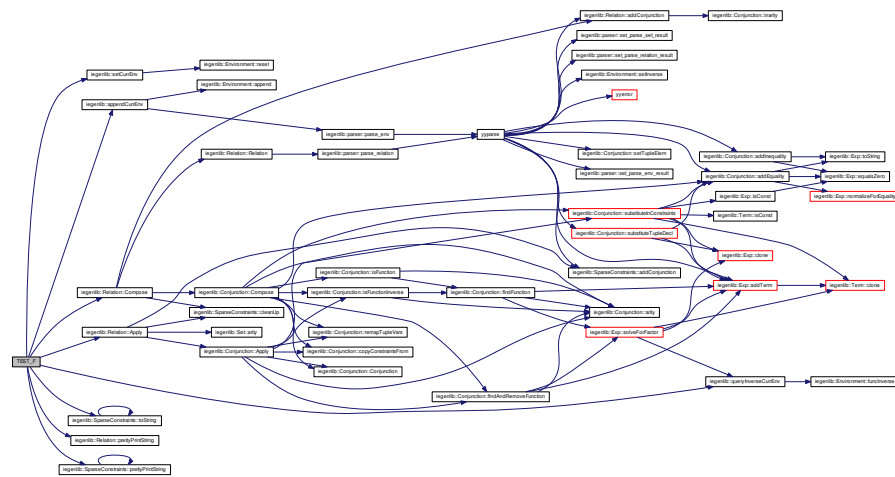
10.16.2.43 TEST\_F( SetRelationTest , IEGRONE41Zeros )

Here is the call graph for this function:



10.16.2.44 TEST\_F( SetRelationTest , IEGRONE72CollapsingInverseFuncs )

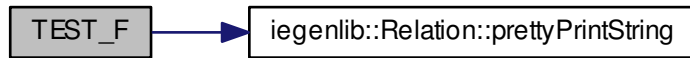
Here is the call graph for this function:





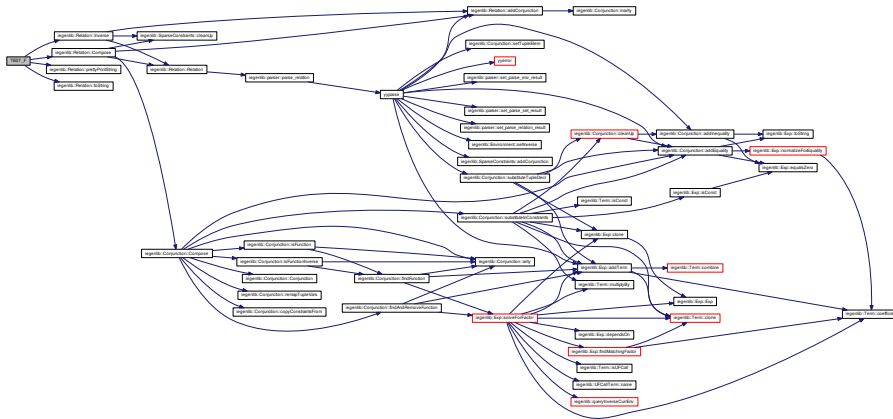
10.16.2.45 TEST\_F ( SetRelationTest , MoldynFSTExample )

Here is the call graph for this function:



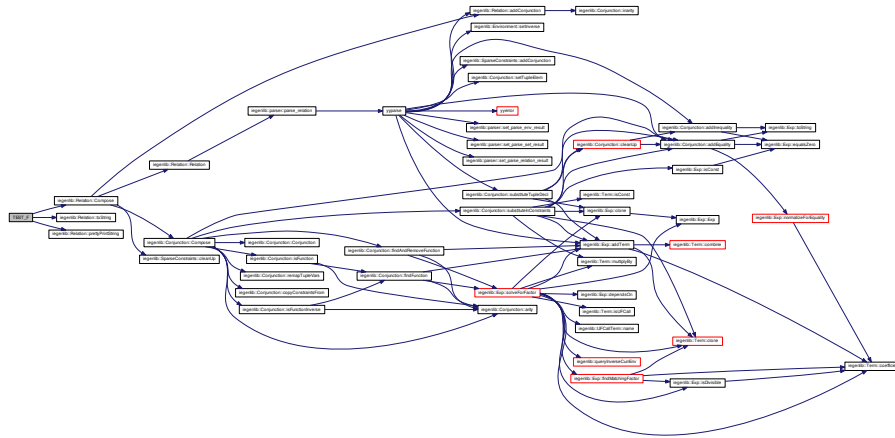
10.16.2.46 TEST\_F ( SetRelationTest , MoldynFSTExampleInverse )

Here is the call graph for this function:



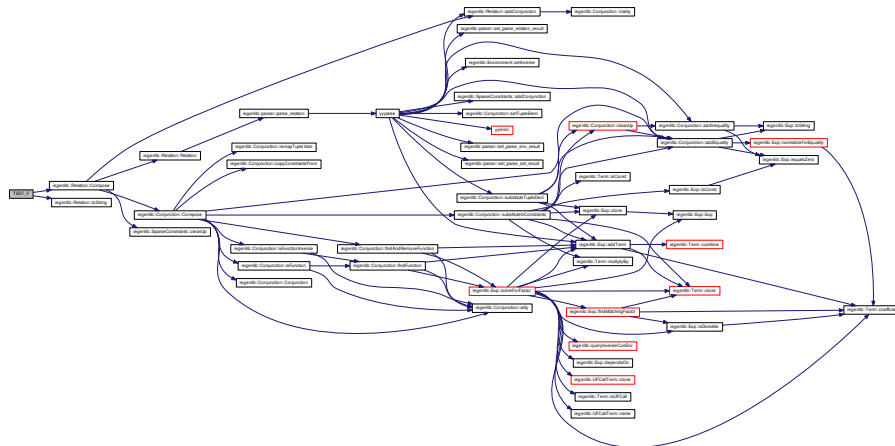
10.16.2.47 TEST\_F ( SetRelationTest , AnandComposeExample )

Here is the call graph for this function:



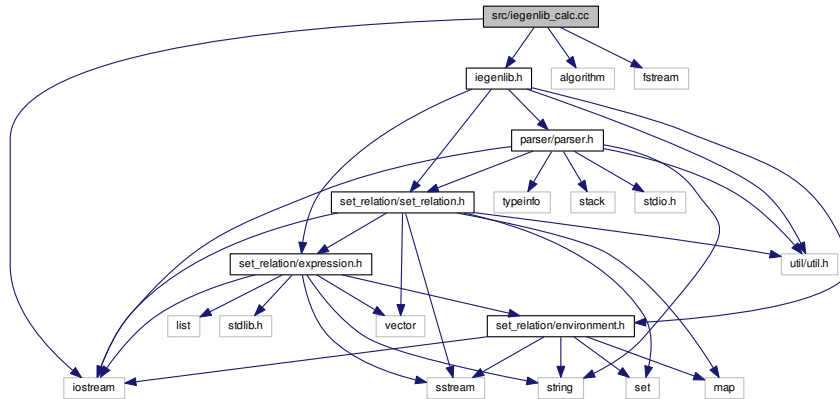
10.16.2.48 TEST\_F ( SetRelationTest , MoldynManyTests )

Here is the call graph for this function:



## 10.17 src/iegenlib\_calc.cc File Reference

```
#include <iostream> #include <algorithm> #include <fstream> ×
#include <iegenlib.h> Include dependency graph for iegenlib_calc.cc:
```



### Defines

- #define [SET\\_RELATION\\_DOT\\_FILE\\_NAME](#) "set\_relation.dot"
- #define [AST\\_DOT\\_FILE\\_NAME](#) "ast.dot"

### Functions

- std::string [lowercase](#) (const std::string &str)
- bool [contains](#) (std::string s, std::string substring, bool caseSensitive=false)
- bool [isRelation](#) (std::string input)
- [Relation](#) \* [newRelationOrNull](#) (std::string input)
- [Set](#) \* [newSetOrNull](#) (std::string input)
- void [handleSet](#) (std::string set\_str)
- void [handleRelation](#) (std::string relation\_str)
- void [handleInverse](#) (std::string set\_relation\_str)
- void [handleCompose](#) (std::string set\_relation\_str)
- void [handleApply](#) (std::string set\_relation\_str)
- void [handleUnion](#) (std::string set\_relation\_str)
- int [main](#) (int ac, char \*\*av)

### 10.17.1 Define Documentation

10.17.1.1 `#define AST_DOT_FILE_NAME "ast.dot"`

10.17.1.2 `#define SET_RELATION_DOT_FILE_NAME "set_relation.dot"`

### 10.17.2 Function Documentation

10.17.2.1 `bool contains ( std::string s, std::string substring, bool caseSensitive = false )`

contains

This is a small helper that reports whether one string contains a substring of interest. It is preferred over directly using the `s.find() != string::npos` idiom because (1) it is shorter, and (2) it more clearly expresses the intent to the human reader. Also, this method can now act in a case-insensitive manner, which `find` doesn't naturally do.

#### Parameters

<code>s</code>	-- string to search
<code>substring</code>	-- substring to search for
<code>case-Sensitive</code>	-- if true, do a case-sensitive search (defaults to false)

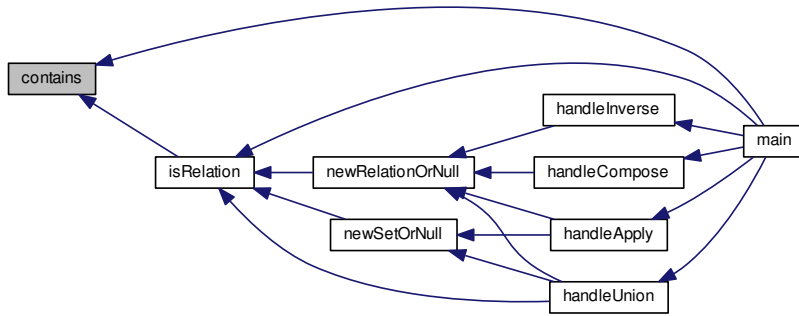
#### Returns

true if s contains substring, false otherwise

Here is the call graph for this function:



Here is the caller graph for this function:

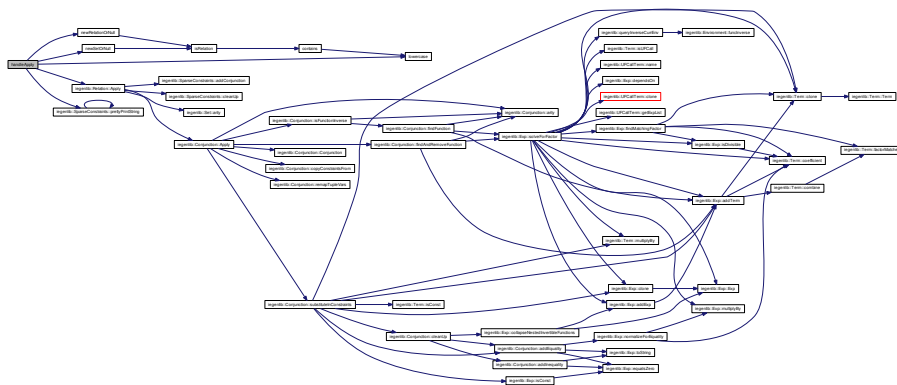


10.17.2.2 void handleApply ( std::string set\_relation\_str )

handleApply

Handle the case of an Apply input.

Here is the call graph for this function:



Here is the caller graph for this function:

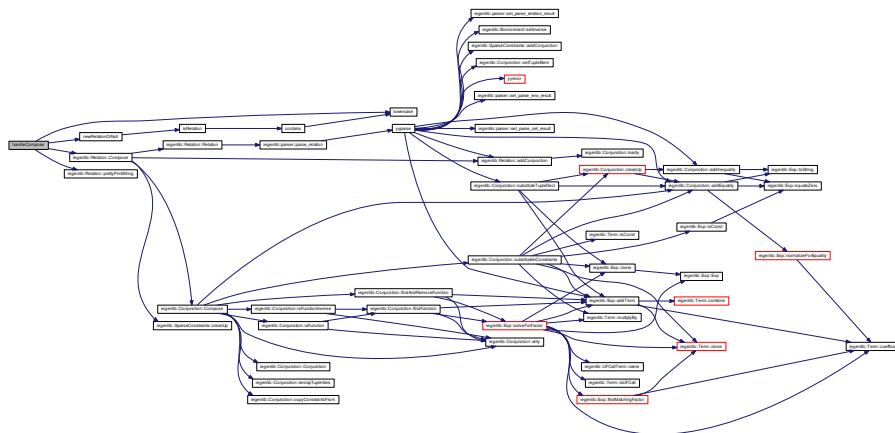


### 10.17.2.3 void handleCompose ( std::string set\_relation\_str )

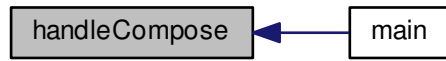
handleCompose

Handle the case of a Compose input.

Here is the call graph for this function:



Here is the caller graph for this function:

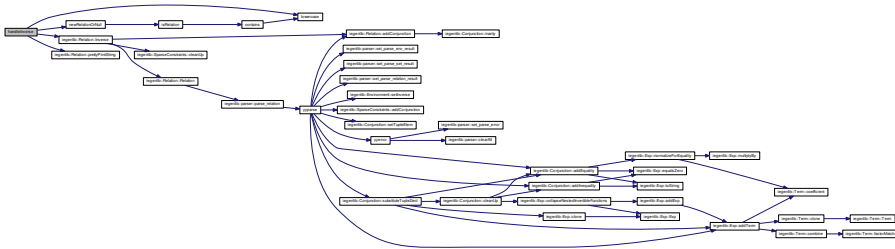


10.17.2.4 void handleInverse ( std::string set\_relation\_str )

handleInverse

Handle the case of an Inverse input.

Here is the call graph for this function:



Here is the caller graph for this function:

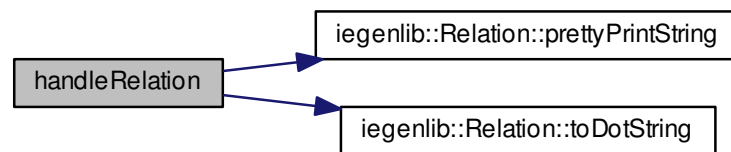


### 10.17.2.5 void handleRelation ( std::string *relation\_str* )

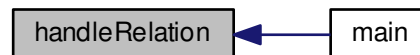
handleRelation

Handle the case of a relation input.

Here is the call graph for this function:



Here is the caller graph for this function:



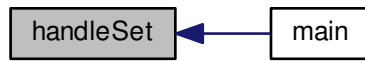
### 10.17.2.6 void handleSet ( std::string *set\_str* )

handleSet

Handle the case of a set input.



Here is the caller graph for this function:

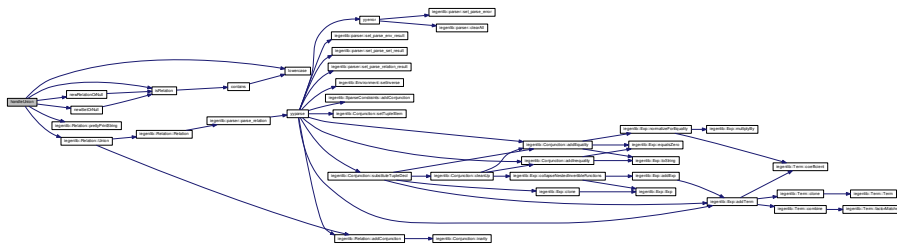


10.17.2.7 void handleUnion ( std::string set\_relation\_str )

handleUnion

Handle the case of a Union input.

Here is the call graph for this function:



Here is the caller graph for this function:



10.17.2.8 `bool isRelation ( std::string input )`

`isRelation`

Determine whether the given input is a relation, by searching for the arrow token (" $\rightarrow$ "), which appears only in a relation (and never in a set).

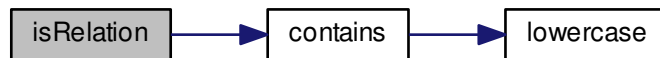
Parameters

<code>input</code>	-- string to examine
--------------------	----------------------

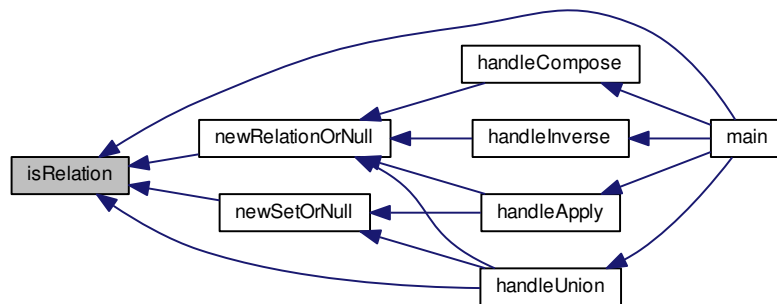
Returns

true iff input appears to be a relation

Here is the call graph for this function:



Here is the caller graph for this function:



## 10.17.2.9 std::string lowercase ( const std::string &amp; str )

lowercase

Converts the given string to lowercase.

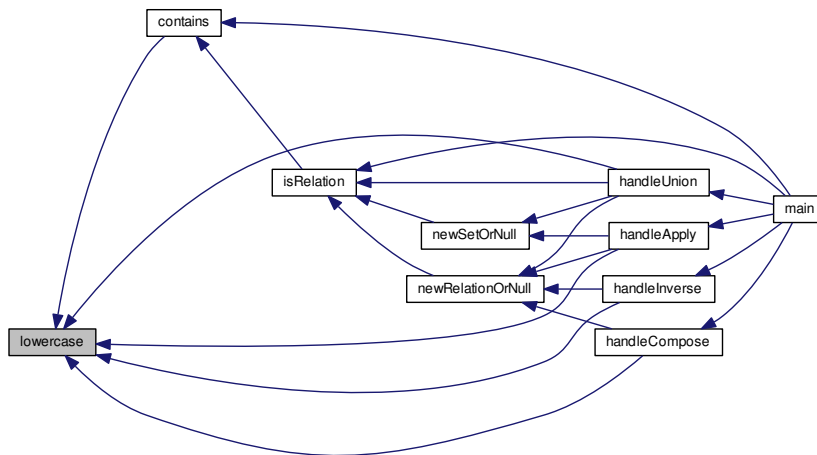
**Parameters**

<i>str</i>	-- string to convert
------------	----------------------

**Returns**

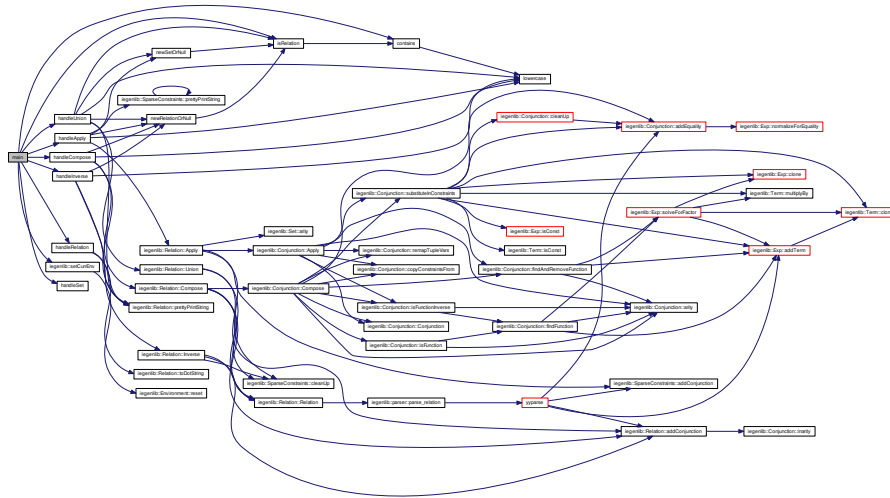
lowercase version of str

Here is the caller graph for this function:



10.17.2.10 `int main ( int ac, char ** av )`

Here is the call graph for this function:

10.17.2.11 `Relation* newRelationOrNull ( std::string input )`

`newRelationOrNull`

This factory function returns a new `Relation` object, if the input string appears to describe a relation; otherwise it returns `NULL`.

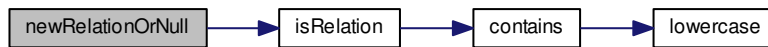
Parameters

<code>input</code> -- string to examine
---

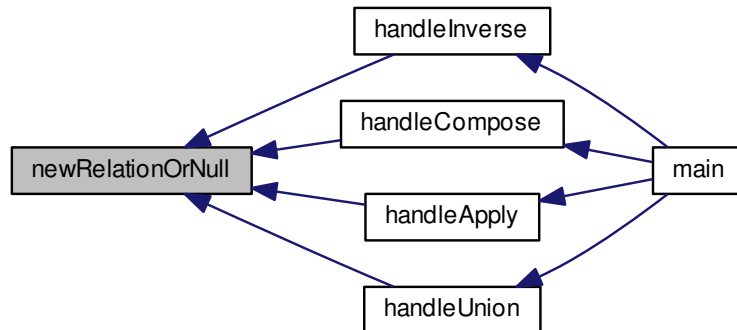
**Returns**

new Relation constructed with input, or NULL

Here is the call graph for this function:



Here is the caller graph for this function:

**10.17.2.12 Set\* newSetOrNull ( std::string input )****newSetOrNull**

This factory function returns a new Set object, if the input string appears to describe a Set; otherwise it returns NULL.

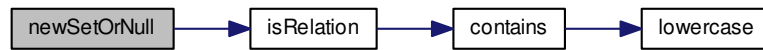
**Parameters**

<i>input</i>	-- string to examine
--------------	----------------------

**Returns**

new Set constructed with input, or NULL

Here is the call graph for this function:



Here is the caller graph for this function:

