

magGen

1.0

Generated by Doxygen 1.6.3

4 September of 2010

Contents

1 magGen - Generator of Static Evaluators of Multi-plans Attribute Grammars.	1
1.1 Introducción	2
1.2 Notas	2
1.3 Requerimientos	2
2 Namespace Index	3
2.1 Namespace List	3
3 Class Hierarchy	5
3.1 Class Hierarchy	5
4 Class Index	7
4.1 Class List	7
5 File Index	9
5.1 File List	9
6 Namespace Documentation	11
6.1 genevalmag Namespace Reference	11
6.1.1 Typedef Documentation	15
6.1.2 Enumeration Type Documentation	16
6.1.3 Function Documentation	17
6.1.4 Variable Documentation	52
6.2 utilities Namespace Reference	54
6.2.1 Function Documentation	54
7 Class Documentation	63
7.1 genevalmag::Attr_grammar Class Reference	63
7.1.1 Detailed Description	65
7.1.2 Constructor & Destructor Documentation	65

7.1.3 Member Function Documentation	66
7.1.4 Member Data Documentation	78
7.2 genevalmag::Attribute Class Reference	80
7.2.1 Detailed Description	81
7.2.2 Constructor & Destructor Documentation	81
7.2.3 Member Function Documentation	82
7.2.4 Member Data Documentation	89
7.3 genevalmag::attribute_grammar Struct Reference	91
7.3.1 Detailed Description	91
7.4 genevalmag::attribute_grammar::definition< ScannerT > Struct Template Reference	92
7.4.1 Detailed Description	93
7.4.2 Member Typedef Documentation	93
7.4.3 Constructor & Destructor Documentation	93
7.4.4 Member Function Documentation	97
7.4.5 Member Data Documentation	97
7.5 genevalmag::Builder_code Class Reference	105
7.5.1 Detailed Description	108
7.5.2 Constructor & Destructor Documentation	108
7.5.3 Member Function Documentation	108
7.5.4 Member Data Documentation	120
7.6 genevalmag::Builder_graphs Class Reference	122
7.6.1 Detailed Description	123
7.6.2 Constructor & Destructor Documentation	124
7.6.3 Member Function Documentation	124
7.6.4 Member Data Documentation	136
7.7 genevalmag::Builder_plans Class Reference	138
7.7.1 Detailed Description	139
7.7.2 Constructor & Destructor Documentation	140
7.7.3 Member Function Documentation	140
7.7.4 Member Data Documentation	154
7.8 genevalmag::Builder_visit_sequences Class Reference	156
7.8.1 Detailed Description	158
7.8.2 Constructor & Destructor Documentation	158
7.8.3 Member Function Documentation	159
7.8.4 Member Data Documentation	162
7.9 genevalmag::c_rule Struct Reference	164

7.9.1 Detailed Description	164
7.9.2 Member Function Documentation	164
7.9.3 Member Data Documentation	165
7.10 genevalmag::cycle_detector Struct Reference	166
7.10.1 Detailed Description	166
7.10.2 Constructor & Destructor Documentation	166
7.10.3 Member Function Documentation	166
7.10.4 Member Data Documentation	167
7.11 genevalmag::decl_attribute Struct Reference	168
7.11.1 Detailed Description	168
7.11.2 Member Data Documentation	168
7.12 genevalmag::Equation Class Reference	169
7.12.1 Detailed Description	171
7.12.2 Constructor & Destructor Documentation	171
7.12.3 Member Function Documentation	172
7.12.4 Member Data Documentation	176
7.13 genevalmag::Expr_function Class Reference	178
7.13.1 Detailed Description	180
7.13.2 Constructor & Destructor Documentation	180
7.13.3 Member Function Documentation	181
7.13.4 Member Data Documentation	190
7.14 genevalmag::Expr_instance Class Reference	191
7.14.1 Detailed Description	193
7.14.2 Constructor & Destructor Documentation	193
7.14.3 Member Function Documentation	195
7.14.4 Member Data Documentation	201
7.15 genevalmag::Expr_leaf Class Reference	203
7.15.1 Detailed Description	204
7.15.2 Constructor & Destructor Documentation	205
7.15.3 Member Function Documentation	205
7.16 genevalmag::Expr_literal Class Reference	207
7.16.1 Detailed Description	209
7.16.2 Constructor & Destructor Documentation	209
7.16.3 Member Function Documentation	210
7.16.4 Member Data Documentation	214
7.17 genevalmag::Expr_node Class Reference	216

7.17.1 Detailed Description	218
7.17.2 Constructor & Destructor Documentation	218
7.17.3 Member Function Documentation	218
7.17.4 Member Data Documentation	220
7.18 genevalmag::Expression Class Reference	221
7.18.1 Detailed Description	224
7.18.2 Constructor & Destructor Documentation	224
7.18.3 Member Function Documentation	224
7.18.4 Member Data Documentation	228
7.19 genevalmag::Function Class Reference	229
7.19.1 Detailed Description	230
7.19.2 Constructor & Destructor Documentation	231
7.19.3 Member Function Documentation	232
7.19.4 Member Data Documentation	243
7.20 genevalmag::i_w Struct Reference	245
7.20.1 Detailed Description	246
7.20.2 Member Function Documentation	246
7.20.3 Member Data Documentation	246
7.21 genevalmag::k_p_project Struct Reference	248
7.21.1 Detailed Description	249
7.21.2 Member Function Documentation	249
7.21.3 Member Data Documentation	250
7.22 genevalmag::k_plan Struct Reference	251
7.22.1 Detailed Description	251
7.22.2 Member Function Documentation	251
7.22.3 Member Data Documentation	252
7.23 genevalmag::k_w Struct Reference	253
7.23.1 Detailed Description	253
7.23.2 Member Function Documentation	253
7.23.3 Member Data Documentation	254
7.24 genevalmag::Maglib Class Reference	255
7.24.1 Detailed Description	255
7.24.2 Constructor & Destructor Documentation	255
7.24.3 Member Function Documentation	255
7.25 genevalmag::Parser_AG Class Reference	257
7.25.1 Detailed Description	258

7.25.2 Constructor & Destructor Documentation	258
7.25.3 Member Function Documentation	258
7.25.4 Member Data Documentation	260
7.26 genevalmag::Rule Class Reference	261
7.26.1 Detailed Description	262
7.26.2 Constructor & Destructor Documentation	262
7.26.3 Member Function Documentation	263
7.26.4 Member Data Documentation	274
7.27 genevalmag::Semantics_checks Class Reference	276
7.27.1 Detailed Description	276
7.27.2 Constructor & Destructor Documentation	276
7.27.3 Member Function Documentation	277
7.27.4 Member Data Documentation	283
7.28 genevalmag::skip_parser Struct Reference	285
7.28.1 Detailed Description	285
7.29 genevalmag::skip_parser::definition< ScannerT > Struct Template Reference . .	286
7.29.1 Detailed Description	286
7.29.2 Constructor & Destructor Documentation	286
7.29.3 Member Function Documentation	286
7.29.4 Member Data Documentation	286
7.30 genevalmag::Sort Class Reference	288
7.30.1 Detailed Description	288
7.30.2 Constructor & Destructor Documentation	288
7.30.3 Member Function Documentation	290
7.30.4 Member Data Documentation	294
7.31 genevalmag::Symbol Class Reference	295
7.31.1 Detailed Description	295
7.31.2 Constructor & Destructor Documentation	296
7.31.3 Member Function Documentation	297
7.31.4 Member Data Documentation	305
7.32 genevalmag::vertex_data_t Struct Reference	306
7.32.1 Detailed Description	306
7.32.2 Member Typedef Documentation	306
8 File Documentation	307
8.1 include/Attr_grammar/Attr_grammar.h File Reference	307
8.1.1 Detailed Description	308

8.2 include/Attr_grammar/Attribute.h File Reference	309
8.2.1 Detailed Description	309
8.3 include/Attr_grammar/Equation.h File Reference	311
8.3.1 Detailed Description	311
8.4 include/Attr_grammar/Function.h File Reference	312
8.4.1 Detailed Description	312
8.5 include/Attr_grammar/Rule.h File Reference	314
8.5.1 Detailed Description	314
8.6 include/Attr_grammar/Sort.h File Reference	316
8.6.1 Detailed Description	316
8.7 include/Attr_grammar/Symbol.h File Reference	317
8.7.1 Detailed Description	317
8.8 include/Builders/Builder_code.h File Reference	319
8.8.1 Detailed Description	320
8.9 include/Builders/Builder_graphs.h File Reference	321
8.9.1 Detailed Description	322
8.10 include/Builders/Builder_plans.h File Reference	323
8.10.1 Detailed Description	324
8.11 include/Builders/Builder_visit_sequences.h File Reference	325
8.11.1 Detailed Description	325
8.12 include/Expression_tree/Expr_function.h File Reference	327
8.12.1 Detailed Description	327
8.13 include/Expression_tree/Expr_instance.h File Reference	328
8.13.1 Detailed Description	328
8.14 include/Expression_tree/Expr_leaf.h File Reference	329
8.14.1 Detailed Description	329
8.15 include/Expression_tree/Expr_literal.h File Reference	330
8.15.1 Detailed Description	330
8.16 include/Expression_tree/Expr_node.h File Reference	331
8.16.1 Detailed Description	331
8.17 include/Expression_tree/Expression.h File Reference	332
8.17.1 Detailed Description	332
8.18 include/Maglib.h File Reference	333
8.18.1 Detailed Description	333
8.19 include/Parser/Parser_AG.h File Reference	334
8.20 include/Parser/Semantics_actions.h File Reference	335

8.20.1 Detailed Description	336
8.21 include/Parser/Semantics_checks.h File Reference	337
8.21.1 Detailed Description	337
8.22 include/Util/Utilities.h File Reference	338
8.22.1 Detailed Description	338
8.23 src/Attr_grammar/Attr_grammar.cpp File Reference	340
8.23.1 Detailed Description	340
8.24 src/Attr_grammar/Attribute.cpp File Reference	341
8.24.1 Detailed Description	341
8.25 src/Attr_grammar/Equation.cpp File Reference	342
8.25.1 Detailed Description	342
8.26 src/Attr_grammar/Function.cpp File Reference	343
8.26.1 Detailed Description	343
8.27 src/Attr_grammar/Rule.cpp File Reference	344
8.27.1 Detailed Description	344
8.28 src/Attr_grammar/Sort.cpp File Reference	345
8.28.1 Detailed Description	345
8.29 src/Attr_grammar/Symbol.cpp File Reference	346
8.29.1 Detailed Description	346
8.30 src/Builders/Builder_code.cpp File Reference	347
8.30.1 Detailed Description	348
8.31 src/Builders/Builder_graphs.cpp File Reference	349
8.31.1 Detailed Description	350
8.32 src/Builders/Builder_plans.cpp File Reference	351
8.32.1 Detailed Description	351
8.33 src/Builders/Builder_visit_sequences.cpp File Reference	352
8.34 src/Expression_tree/Expr_function.cpp File Reference	353
8.34.1 Detailed Description	353
8.35 src/Expression_tree/Expr_instance.cpp File Reference	354
8.35.1 Detailed Description	354
8.36 src/Expression_tree/Expr_leaf.cpp File Reference	355
8.36.1 Detailed Description	355
8.37 src/Expression_tree/Expr_literal.cpp File Reference	356
8.37.1 Detailed Description	356
8.38 src/Expression_tree/Expr_node.cpp File Reference	357
8.38.1 Detailed Description	357

8.39 src/Expression_tree/Expression.cpp File Reference	358
8.39.1 Detailed Description	358
8.40 src/maggen.cpp File Reference	359
8.40.1 Detailed Description	360
8.40.2 Function Documentation	360
8.41 src/Maglib.cpp File Reference	361
8.41.1 Detailed Description	361
8.42 src/Parser/Parser_AG.cpp File Reference	362
8.42.1 Detailed Description	363
8.43 src/Parser/Semantics_actions.cpp File Reference	364
8.43.1 Detailed Description	366
8.44 src/Parser/Semantics_checks.cpp File Reference	367
8.44.1 Detailed Description	367
8.45 src/Util/Utilities.cpp File Reference	368

Chapter 1

magGen - Generator of Static Evaluators of Multi-plans Attribute Grammars.

magGen: Generador de evaluadores estáticos para gramáticas de atributos multiplanos.

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

Copyright (C) 2010 Kilmurray - Picco, UNRC

System: magGen

Homepage: <<http://code.google.com/p/genevalmag/>>

Language: C++

Authors: Kilmurray Gerardo Luis

Authors: Picco Gonzalo Martín

E-Mail: gerakilmurray AT gmail.com

E-Mail: gonzalopicco AT gmail.com

magGen is free software: you can redistribute it and/or modify
it under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

magGen is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with magGen. If not, see <<http://www.gnu.org/licenses/>>.

Director: Marcelo Daniel Arroyo

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

Date

Mayo 2010

1.1 Introducción

Trabajo realizado en el marco de la Tesis de Licenciatura en Ciencias de la Computación.

En este repositorio podrá encontrar:

- bin : Contiene el binario de maggen y archivos necesarios para la generación del evaluador.
- doc : Contiene toda la documentación y tutoriales de la herramienta.
- include : Contiene las cabeceras para incluir los archivos que permiten parsing y análisis de MAG, generación de grafos, planes y secuencias de visitas.
- src : Contiene todos los archivos de implementación de la herramienta.
- examples : Contiene ejemplos para ver el funcionamiento de maggen.
- scripts : Contiene scripts en bash para convertir archivos .dot en imágenes PNG.

Enlaces principales: [Genevalmag](#) [GoogleCode](#)

1.2 Notas

En etapa de escritura del informe

1.3 Requerimientos

Para compilar la herramienta magGen, se necesita:

- Boost libraries 1.41
- g++ 4.3.3

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

genevalmag	11
utilities	54

Chapter 3

Class Hierarchy

3.1 Class Hierarchy

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

genevalmag::Attr_grammar	63
genevalmag::Attribute	80
genevalmag::attribute_grammar	91
genevalmag::Builder_code	105
genevalmag::Builder_graphs	122
genevalmag::Builder_plans	138
genevalmag::Builder_visit_sequences	156
genevalmag::c_rule	164
genevalmag::cycle_detector	166
genevalmag::decl_attribute	168
genevalmag::attribute_grammar::definition< ScannerT >	92
genevalmag::skip_parser::definition< ScannerT >	286
genevalmag::Equation	169
genevalmag::Expression	221
genevalmag::Expr_leaf	203
genevalmag::Expr_instance	191
genevalmag::Expr_literal	207
genevalmag::Expr_node	216
genevalmag::Expr_function	178
genevalmag::Function	229
genevalmag::i_w	245
genevalmag::k_p_project	248
genevalmag::k_plan	251
genevalmag::k_w	253
genevalmag::Maglib	255
genevalmag::Parser_AG	257
genevalmag::Rule	261
genevalmag::Semantics_checks	276
genevalmag::skip_parser	285
genevalmag::Sort	288
genevalmag::Symbol	295
genevalmag::vertex_data_t	306

Chapter 4

Class Index

4.1 Class List

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

genevalmag::Attr_grammar	63
genevalmag::Attribute	80
genevalmag::attribute_grammar	91
genevalmag::Builder_code	105
genevalmag::Builder_graphs	122
genevalmag::Builder_plans	138
genevalmag::Builder_visit_sequences	156
genevalmag::c_rule	164
genevalmag::cycle_detector	166
genevalmag::decl_attribute	168
genevalmag::attribute_grammar::definition< ScannerT >	92
genevalmag::skip_parser::definition< ScannerT >	286
genevalmag::Equation	169
genevalmag::Expr_function	178
genevalmag::Expr_instance	191
genevalmag::Expr_leaf	203
genevalmag::Expr_literal	207
genevalmag::Expr_node	216
genevalmag::Expression	221
genevalmag::Function	229
genevalmag::i_w	245
genevalmag::k_p_project	248
genevalmag::k_plan	251
genevalmag::k_w	253
genevalmag::Maglib	255
genevalmag::Parser_AG	257
genevalmag::Rule	261
genevalmag::Semantics_checks	276
genevalmag::skip_parser	285
genevalmag::Sort	288
genevalmag::Symbol	295
genevalmag::vertex_data_t	306

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

include/Maglib.h (Header of the library of evaluator's generator)	333
include/Attr_grammar/Attr_grammar.h (Class that represent the full attribute grammar)	307
include/Attr_grammar/Attribute.h (Class of the attribute of the attribute grammar)	309
include/Attr_grammar/Equation.h (Class of the equations of the attribute grammar)	311
include/Attr_grammar/Function.h (Class of the function of the attribute grammar)	312
include/Attr_grammar/Rule.h (Class of the rule of the attribute grammar)	314
include/Attr_grammar/Sort.h (Class of the sort of the attribute grammar)	316
include/Attr_grammar/Symbol.h (Class of the symbol of the attribute grammar)	317
include/Builders/Builder_code.h (Header with the functions for build the header and source code of the evaluator of MAG)	319
include/Builders/Builder_graphs.h (Header with the functions for build all kind of graphs for the attribute grammar)	321
include/Builders/Builder_plans.h (Header of generation plans module of Attribute Grammar)	323
include/Builders/Builder_visit_sequences.h (Header with the functions for build the visits sequences of grammar)	325
include/Expression_tree/Expr_function.h (Function element of an Expression)	327
include/Expression_tree/Expr_instance.h (Instance element of an Expression)	328
include/Expression_tree/Expr_leaf.h (Abstract final element of an Expression)	329
include/Expression_tree/Expr_literal.h (Literal element of an Expression)	330
include/Expression_tree/Expr_node.h (Abstract recursive element of an Expression)	331
include/Expression_tree/Expression.h (Abstract expression)	332
include/Parser/Parser_AG.h	334
include/Parser/Semantics_actions.h (Header semantics actions for parse of Attribute grammar)	335
include/Parser/Semantics_checks.h (Header method Semantics checks of Attribute grammar)	337
include/Util/Utilities.h (Header of utilities module, where are methods and function used by many class)	338
src/maggen.cpp (Tool for generation of Static Evaluators for Multiplans Attribute Grammar.)	359
src/Maglib.cpp (Implementation of the library of evaluator's generator)	361

src/Attr_grammar/ Attr_grammar.cpp (Implementation of the methods the Attr_grammar.h)	340
src/Attr_grammar/ Attribute.cpp (Implementation of the methods the Attribute.h)	341
src/Attr_grammar/ Equation.cpp (Implementation of the methods the Equation.h)	342
src/Attr_grammar/ Function.cpp (Implementation of the methods the Function.h)	343
src/Attr_grammar/ Rule.cpp (Implementation of the methods the Rule.h)	344
src/Attr_grammar/ Sort.cpp (Implementation of the methods the Sort.h)	345
src/Attr_grammar/ Symbol.cpp (Implementation of the methods the Symbol.h)	346
src/Builders/ Builder_code.cpp (Implementation of the methods the Builder_code.h)	347
src/Builders/ Builder_graphs.cpp (Implementation of the methods the Builder_graph.h)	349
src/Builders/ Builder_plans.cpp (Implementation of the methods the Builder_plans.h)	351
src/Builders/ Builder_visit_sequences.cpp	352
src/Expression_tree/ Expr_function.cpp (Implementation of a function element of an Expression)	353
src/Expression_tree/ Expr_instance.cpp (Implementation of a instance element of an Expression)	354
src/Expression_tree/ Expr_leaf.cpp (Implementation element of an Expression)	355
src/Expression_tree/ Expr_literal.cpp (Implementation of a literal element of an Expression)	356
src/Expression_tree/ Expr_node.cpp (Implementation element of an Expression)	357
src/Expression_tree/ Expression.cpp (Implementation of an Expression)	358
src/Parser/ Parser_AG.cpp (Header of parsing module of Attribute Grammar)	362
src/Parser/ Semantics_actions.cpp (Implementation of the methods the Semantics_actions.h)	364
src/Parser/ Semantics_checks.cpp (Implementation of the methods the Semantics_checks.h)	367
src/Util/ Utilities.cpp	368

Chapter 6

Namespace Documentation

6.1 genevalmag Namespace Reference

Classes

- class Attr_grammar
- class Attribute
- class Equation
- class Function
- class Rule
- class Sort
- class Symbol
- class Builder_code
- struct vertex_data_t
- class Builder_graphs
- struct c_rule
- struct k_w
- struct i_w
- struct k_plan
- struct k_p_project
- class Builder_plans
- class Builder_visit_sequences
- class Expr_function
- class Expr_instance
- class Expr_leaf
- class Expr_literal
- class Expr_node
- class Expression
- class Maglib
- class Parser_AG
- class Semantics_checks
- struct cycle_detector
- struct skip_parser
- struct attribute_grammar
- struct decl_attribute

Typedefs

- `typedef adjacency_list< hash_setS, vecS, directedS, property_vertex_dp > Graph`
- `typedef Graph::vertex_descriptor Vertex`
- `typedef vector< unsigned short > Order_eval_eq`
- `typedef vector< unsigned short > Order_rule`
- `typedef struct genevalmag::c_rule Context_rule`
- `typedef struct genevalmag::k_w Key_work_list`
- `typedef struct genevalmag::i_w Item_work`
- `typedef struct genevalmag::k_plan Key_plan`
- `typedef struct genevalmag::k_p_project Key_plan_project`
- `typedef vector< int > Visit_seq`
- `typedef char char_t`
- `typedef file_iterator< char_t > iterator_f`
- `typedef position_iterator< iterator_f > iterator_t`

Enumerations

- `enum type_attr { k_inherit, k_synthetize }`
- `enum oper_mode { k_prefix, k_infix, k_postfix }`
- `enum oper_assoc { k_left, k_right, k_non_assoc }`
- `enum symbol_type { k_terminal, k_non_terminal }`
- `enum literal_type {`
- `k_int, k_float, k_char, k_string,`
- `k_bool }`

Functions

- `const bool IS_OPERATOR (true)`
- `void set_attr_grammar (Attr_grammar *at_grammar)`
- `void set_sem_check (Semantics_checks *s_check)`
- `void create_sort (const iterator_t str, const iterator_t end)`
- `void inic_func (const iterator_t str, const iterator_t end)`
- `void add_function (const iterator_t str, const iterator_t end)`
- `void save_name_func (const iterator_t str, const iterator_t end)`
- `void save_domain_func (const iterator_t str, const iterator_t end)`
- `void save_image_func (const iterator_t str, const iterator_t end)`
- `void add_operator (const iterator_t str, const iterator_t end)`
- `void save_mode_op (const iterator_t str, const iterator_t end)`
- `void save_prec_op (int const prec)`
- `void save_assoc_op (const iterator_t str, const iterator_t end)`
- `void add_attribute (const iterator_t str, const iterator_t end)`
- `void save_sort_attr (const iterator_t str, const iterator_t end)`
- `void save_type_attr (const iterator_t str, const iterator_t end)`
- `void save_member_list_attr (const iterator_t str, const iterator_t end)`
- `void create_attributes (const iterator_t str, const iterator_t end)`
- `void create_new_non_terminal (const iterator_t str, const iterator_t end)`
- `void create_new_terminal (const iterator_t str, const iterator_t end)`

- void `create_rule` (const iterator_t str, const iterator_t end)
- void `save_right_side_rule` (const iterator_t str, const iterator_t end)
- void `create_abbreviated_rule` (const iterator_t str, const iterator_t end)
- void `save_rule` (const iterator_t str, const iterator_t end)
- void `create_instance` (const iterator_t str, const iterator_t end)
- void `save_index_ins` (int const index)
- void `save_attr_ins` (const iterator_t str, const iterator_t end)
- void `create_lit_number` (const iterator_t str, const iterator_t end)
- void `create_lit_ch` (const iterator_t ch, const iterator_t end)
- void `create_lit_str` (const iterator_t str, const iterator_t end)
- void `create_bool` (const iterator_t str, const iterator_t end)
- void `create_function` (const iterator_t str, const iterator_t end)
- void `create_operator` (const iterator_t str, const iterator_t end)
- void `create_equation` (const iterator_t str, const iterator_t end)
- void `save_rvalue` (const iterator_t str, const iterator_t end)
- void `push_mark` (char name)
- void `create_literal_node` (const iterator_t str, const iterator_t end)
- void `create_instance_node` (const iterator_t str, const iterator_t end)
- void `create_func_node` (const iterator_t str, const iterator_t end)
- void `create_root_infix_node` (const iterator_t str, const iterator_t end)
- void `create_root_function_node` (const iterator_t str, const iterator_t end)
- void `create_root_postfix_node` (const iterator_t str, const iterator_t end)
- void `create_root_prefix_node` (const iterator_t str, const iterator_t end)
- void `check_well_defined` (const iterator_t str, const iterator_t end)
- void `increment_level` (char name)
- void `decrement_level` (char name)
- template<class K , class T >
 const bool `add` (const T &elem, map< K, T > &map_elem)
- template<class K , class T >
 const string `to_string_map` (const map< K, T > &map_elem)
- const bool `belong` (const Symbol &symb, const string &exprAttrs)
- template<class T >
 string `write_vector_with_inic` (string &text_buffer, const string name_vec, const size_t index, const vector< T > &vec, const string type_vec, const string type_array)
- string `generate_key_plan` (string &text, const string &n_key, const int &num_key, const Key_plan &k_p)
- string `generate_return_index_context` ()
- string `generate_expr_text` (const Expression *node, const Rule &rule)
- const string `PATH_OUTPUT_GRAPHS` ("graphs/")
- const string `PATH_OUTPUT_DP` ("1_DP_graphs/")
- const string `PATH_OUTPUT_DOWN` ("2_DOWN_graphs/")
- const string `PATH_OUTPUT_DCG` ("3_DCG_graphs/")
- const string `PATH_OUTPUT_ADG` ("4_ADG_graphs/")
- const string `PATH_OUTPUT_CYCLIC` ("CYCLIC_graphs/")
- const string `FILE_DP_GRAPH` ("dp_graph")
- const string `FILE_DOWN_GRAPH` ("down_graph")
- const string `FILE_DCG_GRAPH` ("dcg_graph")
- const string `FILE_ADG_GRAPH` ("adg_graph")
- const string `FILE_ADG_SUBGRAPH_CYCLIC` ("adg_subgraph_with_cyclic")
- const string `PATH_OUT_PLAN` ("plans/")

- const string PATH_OUT_PLAN_PROJECT ("plans_project/")
- void **purge_plan_with** (const Rule &rule, const Order_eval_eq &order_eq, Order_eval_eq &purged_order)
- bool **defined_work** (const vector< Item_work > &list, const Item_work &item_work)
- unsigned short **return_index_vec** (const Order_eval_eq &order, vector< Order_eval_eq > &vec)
- const unsigned short **LEAVE** (0)
- bool **ins_attr_computed** (const Expr_instance *ins, const vector< Expr_instance > &vec)
- void **get_inherits_of** (const Symbol *symb, const vector< Expr_instance > &computed, vector< Expr_instance > &rec_child)
- bool **belong_it** (const map< Key_plan, unsigned short >::const_iterator it, const vector< map< Key_plan, unsigned short >::const_iterator > &vec)
- void **merge_vec** (const vector< map< Key_plan, unsigned short >::const_iterator > &vec_source, vector< map< Key_plan, unsigned short >::const_iterator > &vec_targed)
- void **merge_vec_without_plan** (const vector< map< Key_plan, unsigned short >::const_iterator > &vec_source, vector< map< Key_plan, unsigned short >::const_iterator > &vec_targed, const map< Key_plan, unsigned short >::const_iterator &plan)
- void **plan_family_computed** (const vector< map< Key_plan, unsigned short >::const_iterator > &plans_computed, vector< unsigned short > &visit_seq_computed)
- const string **DEFAULT_PATH** ("./out_maggen/")
- const string **DEFAULT_FILE_NAME** ("mag_eval")
- const string **DEFAULT_INPUT_FILE** ("/tmp/.input_maggen_default")
- double **timeval_diff** (struct timeval *a, struct timeval *b)
- bool **check_file_exist** (const string &strFilename)
- bool **check_name** (const string &strFilename)
- void **show_help_information** ()
- bool **parse_parameters** (int argc, char *argv[], string &path_input_file, string &path_folder_output, string &name_library, vector< string > &headers)
- const string **FILE_GRAMMAR** ("Grammar_mag.log")
- std::ostream & operator<< (std::ostream &out, file_position const &lc)
- int **swap_root_child** (Expr_function **old_root, int i_new_root)
- int **swap_root_grandson** (Expr_function **old_root)
- void **warshall_algorithm** (const unsigned int size, bool *matrix_plain)
- int **get_index** (string name_symb, vector< string > non_term)
- bool **check_eqDefines_it** (const Symbol *symb, const int index, const Attribute *attr, const map< unsigned short, Equation > eqs)

Variables

- vector< Equation * > index_access_eq
- Attr_grammar * attr_grammar
- Semantics_checks * sem_check
- Function * current_func
- struct genevalmag::decl_attribute * newAttrs
- Rule * current_rule
- Expr_instance * current_instance
- Expr_literal * current_literal
- Expr_function * current_ast_function
- Equation * current_eq
- vector< Expression * > stack_node
- vector< Expr_node * > stack_inner_node

6.1.1 Typedef Documentation

6.1.1.1 **typedef char genevalmag::char_t**

Type definitions to report parsing errors, showing row and column.

Definition at line 27 of file Parser_AG.h.

6.1.1.2 **typedef struct genevalmag::c_rule genevalmag::Context_rule**

This struct represent a Context_rule, that is a father rule and the context of rule.

6.1.1.3 **typedef adjacency_list<hash_setS, vecS, directedS, property_vertex_dp> genevalmag::Graph**

Definition at line 35 of file Builder_graphs.h.

6.1.1.4 **typedef struct genevalmag::i_w genevalmag::Item_work**

This struct represents an Item work, that is an Key_work_list and an equations order evaluation.

6.1.1.5 **typedef file_iterator<char_t> genevalmag::iterator_f**

Definition at line 28 of file Parser_AG.h.

6.1.1.6 **typedef position_iterator<iterator_f> genevalmag::iterator_t**

Definition at line 29 of file Parser_AG.h.

6.1.1.7 **typedef struct genevalmag::k_plan genevalmag::Key_plan**

This structs represent a Key_plan, that is an id and an equations order evaluation.

6.1.1.8 **typedef struct genevalmag::k_p_project genevalmag::Key_plan_project**

This struct represent a Key_plan_project, that is a Key_plan, the symbol and occurrence, by which project.

6.1.1.9 **typedef struct genevalmag::k_w genevalmag::Key_work_list**

This struct represents a Key_work_list, that is a father rule and id-rule of the current rule.

6.1.1.10 `typedef vector< unsigned short > genevalmag::Order_eval_eq`

Vector for order of equation. Each number represent an equation.

Definition at line 28 of file Builder_plans.h.

6.1.1.11 `typedef vector< unsigned short > genevalmag::Order_rule`

Vector that represents the inferior context of rule. Each number represent a rule.

Definition at line 34 of file Builder_plans.h.

6.1.1.12 `typedef Graph::vertex_descriptor genevalmag::Vertex`

Definition at line 38 of file Builder_graphs.h.

6.1.1.13 `typedef vector<int> genevalmag::Visit_seq`

Where: item > 0 represent to visit to this rule number. item == 0 represent to leave. item < 0 represent to compute this equation number.

Definition at line 28 of file Builder_visit_sequences.h.

6.1.2 Enumeration Type Documentation

6.1.2.1 `enum genevalmag::literal_type`

Type of basics literals.

Enumerator:

- k_int*
- k_float*
- k_char*
- k_string*
- k_bool*

Definition at line 20 of file Expr_literal.h.

6.1.2.2 `enum genevalmag::oper_assoc`

Enumeration of operation's associative modes.

Enumerator:

- k_left*
- k_right*
- k_non_assoc*

Definition at line 33 of file Function.h.

6.1.2.3 enum genevalmag::oper_mode

Enumeration of operation's mode.

Enumerator:

k_prefix

k_infix

k_postfix

Definition at line 23 of file Function.h.

6.1.2.4 enum genevalmag::symbol_type

Enumeration of symbol's type.

Enumerator:

k_terminal

k_non_terminal

Definition at line 23 of file Symbol.h.

6.1.2.5 enum genevalmag::type_attr

Enumeration of the attribute's type: Inherit or Synthetize.

Enumerator:

k_inherit

k_synthetize

Definition at line 20 of file Attribute.h.

6.1.3 Function Documentation

6.1.3.1 template<class K , class T > const bool genevalmag::add (const T & elem, map< K, T > & map_elem) [inline]

Operation template that insert an object type T in the parameter map.

Returns true if insert successfully. In other case, return false.

Definition at line 42 of file Attr_grammar.cpp.

6.1.3.2 void genevalmag::add_attribute (const iterator_t str, const iterator_t end)

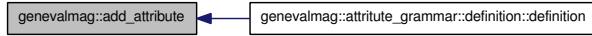
Methods and functions for parse [Attribute](#) class.

Definition at line 161 of file Semantics_actions.cpp.

References `genevalmag::decl_attribute::d_member_symbol`, `genevalmag::decl_attribute::d_mod_type`, `genevalmag::decl_attribute::d_names`, `k_synthetize`, and `new_attrs`.

Referenced by `genevalmag::attritute_grammar::definition< ScannerT >::definition()`.

Here is the caller graph for this function:



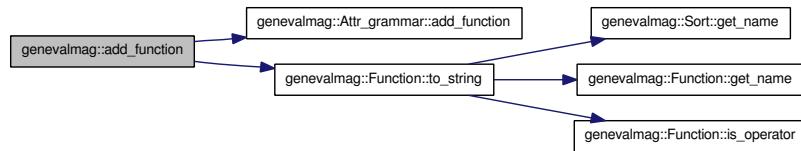
6.1.3.3 void genevalmag::add_function (const iterator_t str, const iterator_t end)

Definition at line 104 of file `Semantics_actions.cpp`.

References `genevalmag::Attr_grammar::add_function()`, `attr_grammar`, `current_func`, and `genevalmag::Function::to_string()`.

Referenced by `add_operator()`, and `genevalmag::attritute_grammar::definition< ScannerT >::definition()`.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.4 void genevalmag::add_operator (const iterator_t str, const iterator_t end)

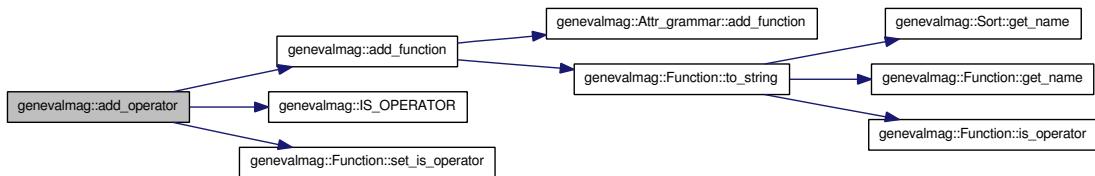
Methods and functions for parse Operator.

Definition at line 135 of file `Semantics_actions.cpp`.

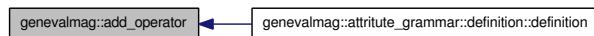
References `add_function()`, `current_func`, `IS_OPERATOR()`, and `genevalmag::Function::set_is_operator()`.

Referenced by `genevalmag::attritute_grammar::definition< ScannerT >::definition()`.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.5 const bool genevalmag::belong (const Symbol & symb, const string & expr_attrs)

Interprets the expression of sets and returns true if the symbol belongs to that set.

Definition at line 94 of file Attr_grammar.cpp.

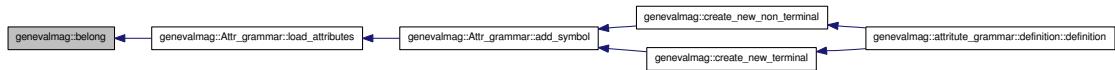
References genevalmag::Symbol::get_name().

Referenced by genevalmag::Attr_grammar::load_attributes().

Here is the call graph for this function:



Here is the caller graph for this function:



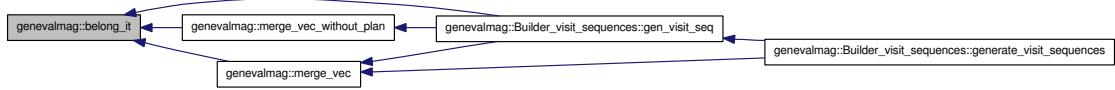
6.1.3.6 bool genevalmag::belong_it (const map< Key_plan, unsigned short >::const_iterator it, const vector< map< Key_plan, unsigned short >::const_iterator > & vec)

Returns true if the iterator belongs to the vector passed as parameter.

Definition at line 78 of file Builder_visit_sequences.cpp.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), merge_vec(), and merge_vec_without_plan().

Here is the caller graph for this function:



6.1.3.7 bool genevalmag::check_eqDefinesIt (const Symbol * symb, const int index, const Attribute * attr, const map< unsigned short, Equation > eqs)

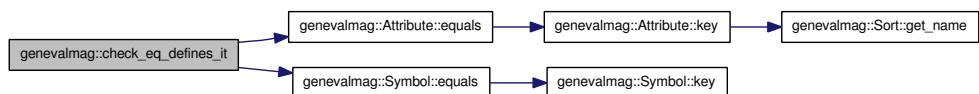
Checks if exist an equation that define the instance formed with the parameters.

Definition at line 469 of file Semantics_checks.cpp.

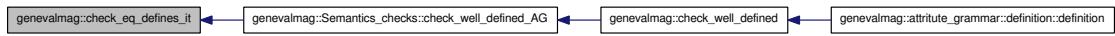
References genevalmag::Attribute::equals(), and genevalmag::Symbol::equals().

Referenced by genevalmag::Semantics_checks::check_well_defined_AG().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.8 bool genevalmag::check_file_exist (const string & strFilename)

Definition at line 116 of file maggen.cpp.

Referenced by parse_parameters().

Here is the caller graph for this function:

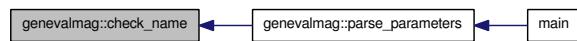


6.1.3.9 bool genevalmag::check_name (const string & strFilename)

Definition at line 128 of file maggen.cpp.

Referenced by parse_parameters().

Here is the caller graph for this function:



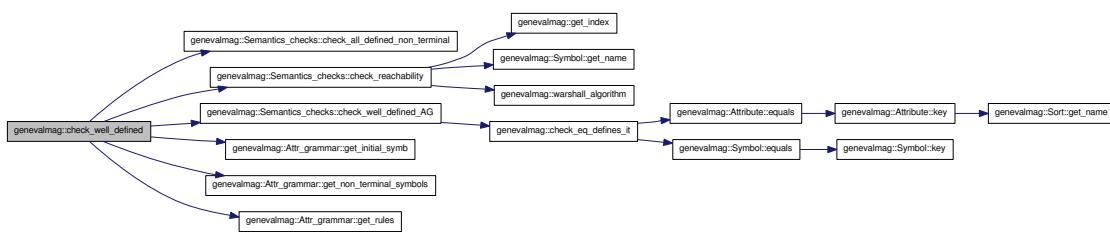
6.1.3.10 void genevalmag::check_well_defined (const iterator_t str, const iterator_t end)

Definition at line 654 of file Semantics_actions.cpp.

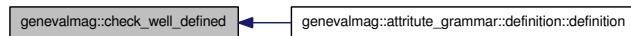
References attr_grammar, genevalmag::Semantics_checks::check_all_defined_non_terminal(), genevalmag::Semantics_checks::check_reachability(), genevalmag::Semantics_checks::check_well_defined_AG(), genevalmag::Attr_grammar::get_initial_symb(), genevalmag::Attr_grammar::get_non_terminal_symbols(), genevalmag::Attr_grammar::get_rules(), and sem_check.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



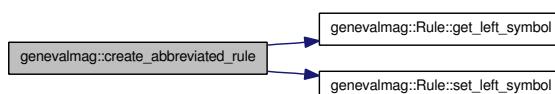
6.1.3.11 void genevalmag::create_abbreviated_rule (const iterator_t str, const iterator_t end)

Definition at line 253 of file Semantics_actions.cpp.

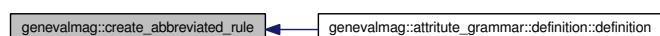
References current_rule, genevalmag::Rule::get_left_symbol(), and genevalmag::Rule::set_left_symbol().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



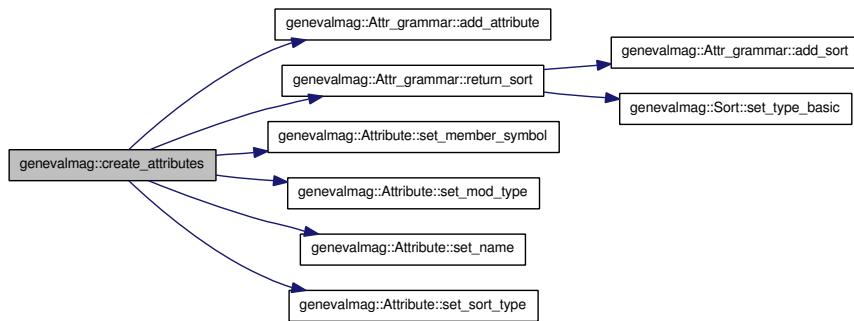
6.1.3.12 void genevalmag::create_attributes (const iterator_t str, const iterator_t end)

Definition at line 204 of file Semantics_actions.cpp.

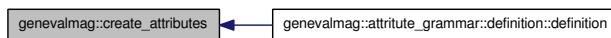
References genevalmag::Attr_grammar::add_attribute(), attr_grammar, genevalmag::decl_attribute::d_member_symbol, genevalmag::decl_attribute::d_mod_type, genevalmag::decl_attribute::d_names, genevalmag::decl_attribute::d_sort_type, newAttrs, genevalmag::Attr_grammar::return_sort(), genevalmag::Attribute::set_member_symbol(), genevalmag::Attribute::set_mod_type(), genevalmag::Attribute::set_name(), and genevalmag::Attribute::set_sort_type().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



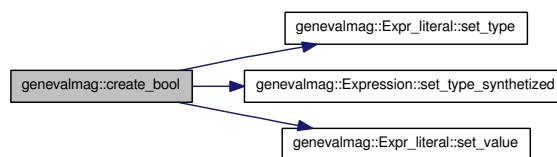
6.1.3.13 void genevalmag::create_bool (const iterator_t str, const iterator_t end)

Definition at line 361 of file Semantics_actions.cpp.

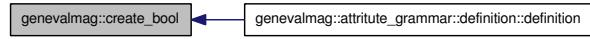
References current_literal, k_bool, genevalmag::Expr_literal::set_type(), genevalmag::Expression::set_type_synthesized(), and genevalmag::Expr_literal::set_value().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.14 void genevalmag::create_equation (const iterator_t str, const iterator_t end)

Definition at line 385 of file Semantics_actions.cpp.

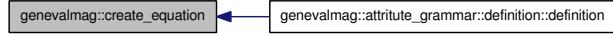
References current_eq, current_instance, and genevalmag::Equation::set_l_value().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



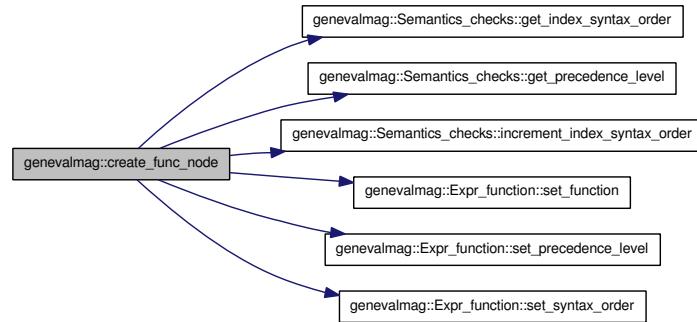
6.1.3.15 void genevalmag::create_func_node (const iterator_t str, const iterator_t end)

Definition at line 459 of file Semantics_actions.cpp.

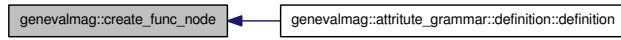
References current_ast_function, current_func, genevalmag::Semantics_checks::get_index_syntax_order(), genevalmag::Semantics_checks::get_precedence_level(), genevalmag::Semantics_checks::increment_index_syntax_order(), sem_check, genevalmag::Expr_function::set_function(), genevalmag::Expr_function::set_precedence_level(), genevalmag::Expr_function::set_syntax_order(), and stack_inner_node.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



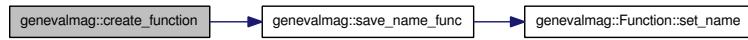
6.1.3.16 void genevalmag::create_function (const iterator_t str, const iterator_t end)

Definition at line 373 of file Semantics_actions.cpp.

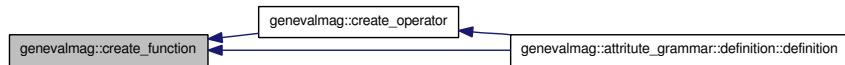
References current_func, and save_name_func().

Referenced by create_operator(), and genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.17 void genevalmag::create_instance (const iterator_t str, const iterator_t end)

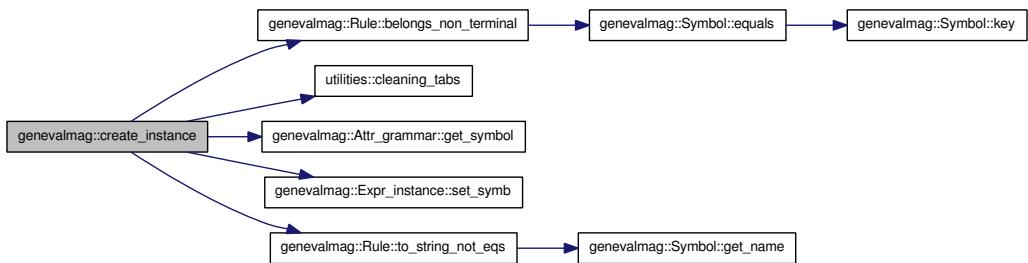
Methods and functions for parse [Equation](#) class of [Rule](#).

Definition at line 268 of file Semantics_actions.cpp.

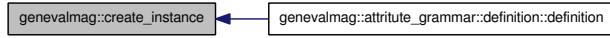
References attr_grammar, genevalmag::Rule::belongs_non_terminal(), utilities::cleaning_tabs(), current_instance, current_rule, genevalmag::Attr_grammar::get_symbol(), genevalmag::Expr_instance::set_symb(), and genevalmag::Rule::to_string_not_eqs().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



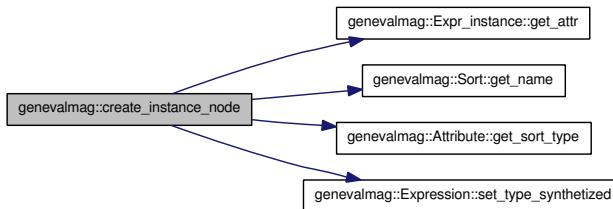
6.1.3.18 void genevalmag::create_instance_node (const iterator_t str, const iterator_t end)

Definition at line 450 of file Semantics_actions.cpp.

References current_instance, genevalmag::Expr_instance::get_attr(), genevalmag::Sort::get_name(), genevalmag::Attribute::get_sort_type(), genevalmag::Expression::set_type_synthetized(), and stack_node.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



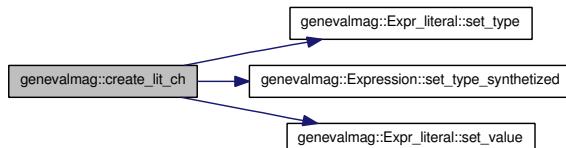
6.1.3.19 void genevalmag::create_lit_ch (const iterator_t ch, const iterator_t end)

Definition at line 333 of file Semantics_actions.cpp.

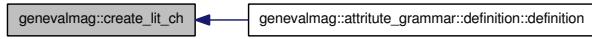
References current_literal, k_char, genevalmag::Expr_literal::set_type(), genevalmag::Expression::set_type_synthetized(), and genevalmag::Expr_literal::set_value().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



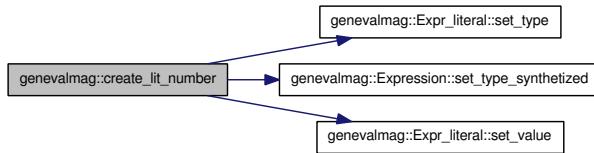
6.1.3.20 void genevalmag::create_lit_number (const iterator_t str, const iterator_t end)

Definition at line 309 of file Semantics_actions.cpp.

References current_literal, k_float, k_int, genevalmag::Expr_literal::set_type(), genevalmag::Expression::set_type_synthesized(), and genevalmag::Expr_literal::set_value().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



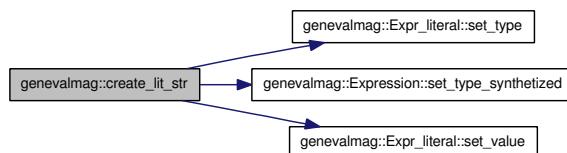
6.1.3.21 void genevalmag::create_lit_str (const iterator_t str, const iterator_t end)

Definition at line 347 of file Semantics_actions.cpp.

References current_literal, k_string, genevalmag::Expr_literal::set_type(), genevalmag::Expression::set_type_synthesized(), and genevalmag::Expr_literal::set_value().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.22 void genevalmag::create_literal_node (const iterator_t str, const iterator_t end)

Creation expression nodes.

Definition at line 443 of file Semantics_actions.cpp.

References current_literal, and stack_node.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the caller graph for this function:



6.1.3.23 void genevalmag::create_new_non_terminal (const iterator_t str, const iterator_t end)

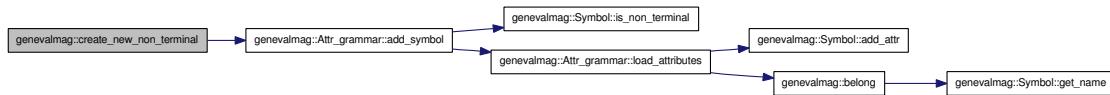
Methods and functions for parse [Symbol](#) class.

Definition at line 223 of file Semantics_actions.cpp.

References genevalmag::Attr_grammar::add_symbol(), attr_grammar, and k_non_terminal.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



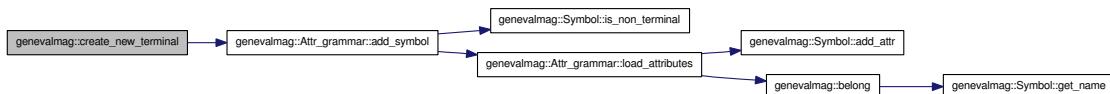
6.1.3.24 void genevalmag::create_new_terminal (const iterator_t str, const iterator_t end)

Definition at line 230 of file Semantics_actions.cpp.

References genevalmag::Attr_grammar::add_symbol(), attr_grammar, and k_terminal.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



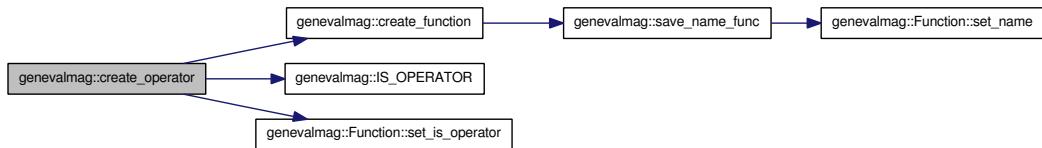
6.1.3.25 void genevalmag::create_operator (const iterator_t str, const iterator_t end)

Definition at line 379 of file Semantics_actions.cpp.

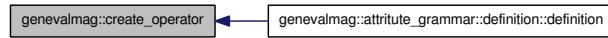
References `create_function()`, `current_func`, `IS_OPERATOR()`, and `genevalmag::Function::set_is_operator()`.

Referenced by `genevalmag::attribute_grammar::definition< ScannerT >::definition()`.

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.26 void genevalmag::create_root_function_node (const iterator_t str, const iterator_t end)

Definition at line 524 of file Semantics_actions.cpp.

References `attr_grammar`, `genevalmag::Attr_grammar::get_function()`, `stack_inner_node`, and `stack_node`.

Referenced by `genevalmag::attribute_grammar::definition< ScannerT >::definition()`.

Here is the call graph for this function:



Here is the caller graph for this function:



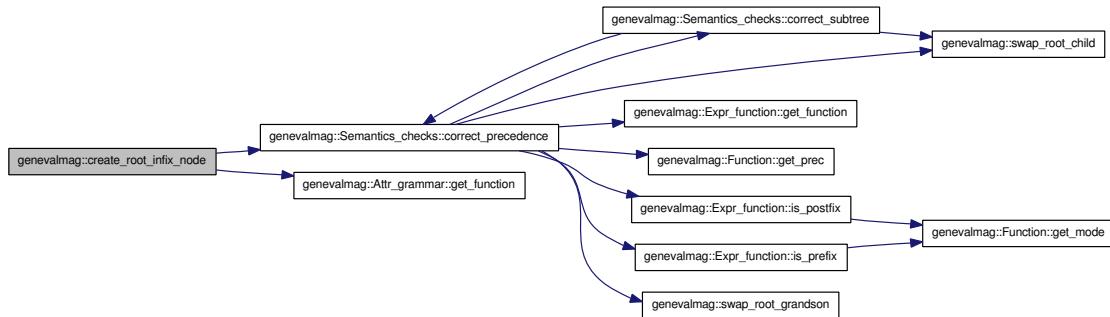
6.1.3.27 void genevalmag::create_root_infix_node (const iterator_t str, const iterator_t end)

Definition at line 477 of file Semantics_actions.cpp.

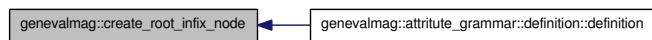
References attr_grammar, genevalmag::Semantics_checks::correct_precedence(), genevalmag::Attr_grammar::get_function(), sem_check, stack_inner_node, and stack_node.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



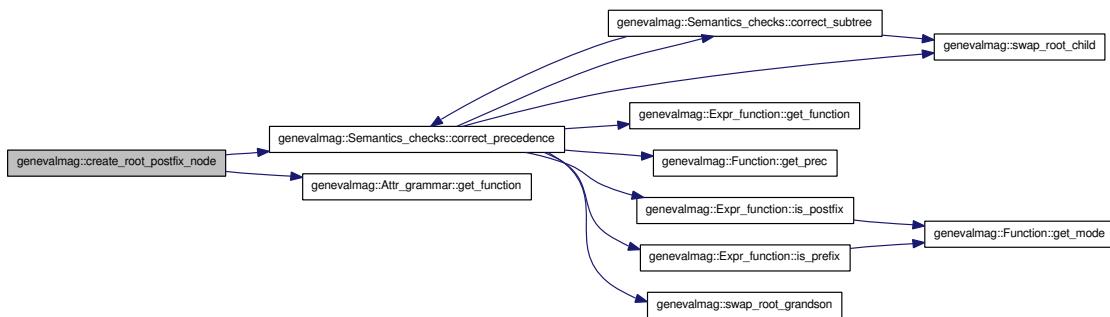
6.1.3.28 void genevalmag::create_root_postfix_node (const iterator_t str, const iterator_t end)

Definition at line 574 of file Semantics_actions.cpp.

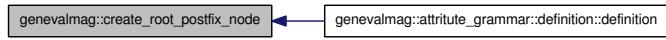
References attr_grammar, genevalmag::Semantics_checks::correct_precedence(), genevalmag::Attr_grammar::get_function(), sem_check, stack_inner_node, and stack_node.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



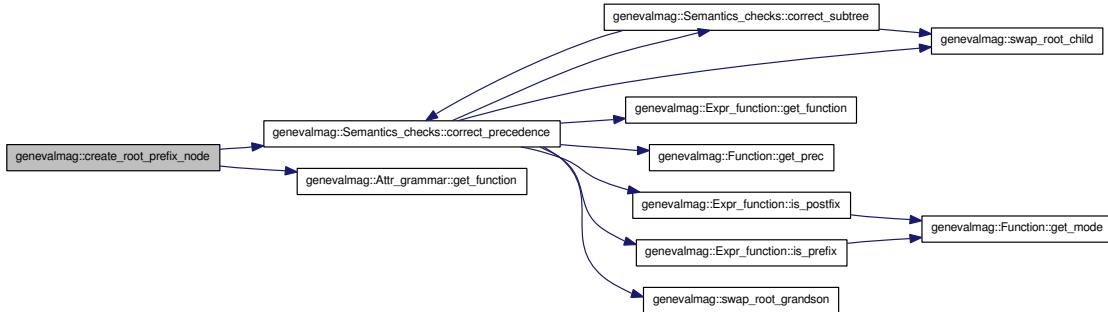
6.1.3.29 void genevalmag::create_root_prefix_node (const iterator_t str, const iterator_t end)

Definition at line 614 of file Semantics_actions.cpp.

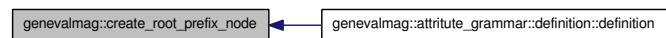
References attr_grammar, genevalmag::Semantics_checks::correct_precedence(), genevalmag::Attr_grammar::get_function(), sem_check, stack_inner_node, and stack_node.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.30 void genevalmag::create_rule (const iterator_t str, const iterator_t end)

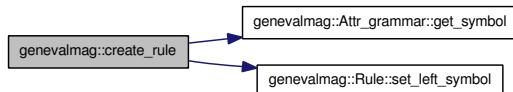
Methods and functions for parse Rule class.

Definition at line 240 of file Semantics_actions.cpp.

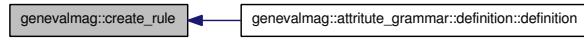
References attr_grammar, current_rule, genevalmag::Attr_grammar::get_symbol(), and genevalmag::Rule::set_left_symbol().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.31 void genevalmag::create_sort (const iterator_t str, const iterator_t end)

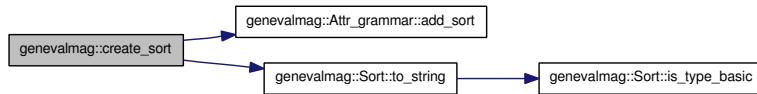
Methods and functions for parse Sort class.

Definition at line 85 of file Semantics_actions.cpp.

References genevalmag::Attr_grammar::add_sort(), attr_grammar, and genevalmag::Sort::to_string().

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



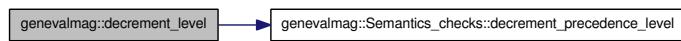
6.1.3.32 void genevalmag::decrement_level (char name)

Definition at line 668 of file Semantics_actions.cpp.

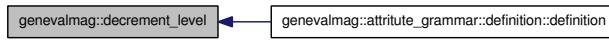
References genevalmag::Semantics_checks::decrement_precedence_level(), and sem_check.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.33 const string genevalmag::DEFAULT_FILE_NAME ("mag_eval")

Default name of the files generated by the library.

Referenced by main().

Here is the caller graph for this function:



6.1.3.34 const string genevalmag::DEFAULT_INPUT_FILE ("~/tmp/.input_maggen_- default")

Default input file.

Referenced by main().

Here is the caller graph for this function:



6.1.3.35 const string genevalmag::DEFAULT_PATH ("./out_maggen/")

Default output path of the generation code.

Referenced by main().

Here is the caller graph for this function:



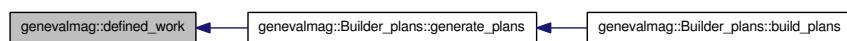
6.1.3.36 bool genevalmag::defined_work (const vector< Item_work > & list, const Item_work & item_work)

Searchs in the list the item work that passed as parameter, if it find return true, otherwise false.

Definition at line 338 of file Builder_plans.cpp.

Referenced by genevalmag::Builder_plans::generate_plans().

Here is the caller graph for this function:



6.1.3.37 const string genevalmag::FILE_ADG_GRAPH ("adp_graph")

Referenced by genevalmag::Builder_graphs::save_adp_graphs().

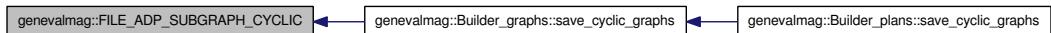
Here is the caller graph for this function:



6.1.3.38 const string genevalmag::FILE_AD_P_SUBGRAPH_CYCLIC ("_adp_subgraph_with_cyclic")

Referenced by genevalmag::Builder_graphs::save_cyclic_graphs().

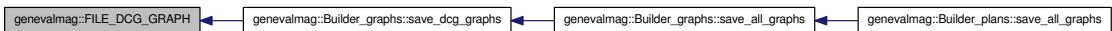
Here is the caller graph for this function:



6.1.3.39 const string genevalmag::FILE_DCG_GRAPH ("_dcg_graph")

Referenced by genevalmag::Builder_graphs::save_dcg_graphs().

Here is the caller graph for this function:



6.1.3.40 const string genevalmag::FILE_DOWN_GRAPH ("_down_graph")

Referenced by genevalmag::Builder_graphs::save_down_graphs().

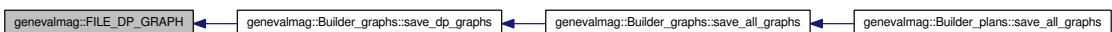
Here is the caller graph for this function:



6.1.3.41 const string genevalmag::FILE_DP_GRAPH ("_dp_graph")

Referenced by genevalmag::Builder_graphs::save_dp_graphs().

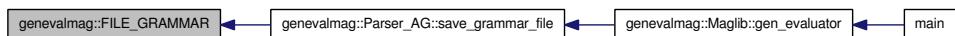
Here is the caller graph for this function:



6.1.3.42 const string genevalmag::FILE_GRAMMAR ("Grammar_mag.log")

Referenced by genevalmag::Parser_AG::save_grammar_file().

Here is the caller graph for this function:



6.1.3.43 string genevalmag::generate_expr_text (const Expression * node, const Rule & rule)

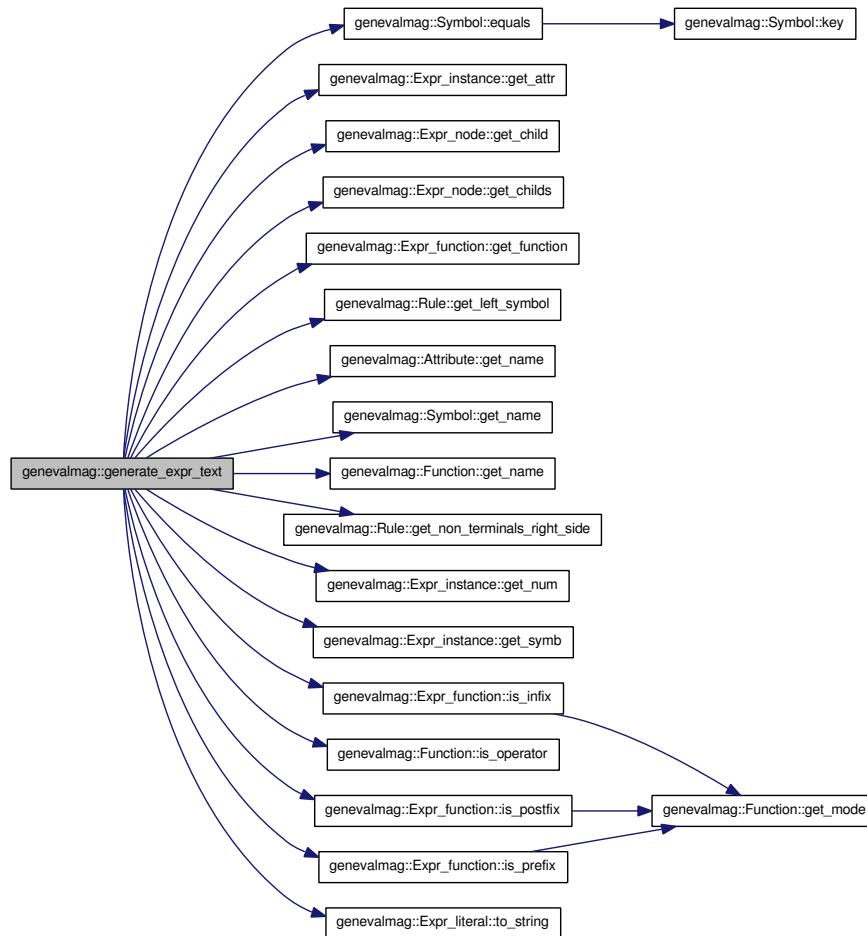
Generates the plain text of a equation of this rule.

Definition at line 643 of file Builder_code.cpp.

References genevalmag::Symbol::equals(), genevalmag::Expr_instance::get_attr(), genevalmag::Expr_node::get_child0(), genevalmag::Expr_function::get_function(), genevalmag::Attribute::get_name(), genevalmag::Function::get_name(), genevalmag::Expr_instance::get_num(), genevalmag::Expr_function::is_infix(), genevalmag::Expr_function::is_postfix(), genevalmag::Expr_literal::to_string(), genevalmag::Expr_node::get_left_symbol(), genevalmag::Rule::get_left_side(), genevalmag::Expr_instance::get_non_terminals_right_side(), genevalmag::Expr_instance::get_symb(), genevalmag::Rule::get_operator(), and genevalmag::Function::is_operator().

Referenced by genevalmag::Builder_code::generate_all_methods_eqs().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.44 `string genevalmag::generate_key_plan (string & text, const string & n_key, const int & num_key, const Key_plan & k_p)`

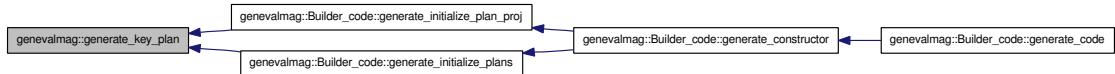
Generate a key plan with the parameters.

Definition at line 306 of file `Builder_code.cpp`.

References `genevalmag::k_plan::id_plan`, and `genevalmag::k_plan::plan`.

Referenced by `genevalmag::Builder_code::generate_initialize_plan_proj0`, and `genevalmag::Builder_code::generate_initialize_plans()`.

Here is the caller graph for this function:



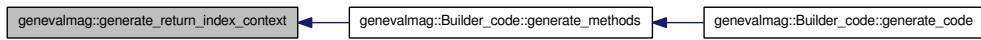
6.1.3.45 string genevalmag::generate_return_index_context ()

Generates the return_index_context method, for get the index of a context rule.

Definition at line 569 of file Builder_code.cpp.

Referenced by genevalmag::Builder_code::generate_methods().

Here is the caller graph for this function:



6.1.3.46 int genevalmag::get_index (string name_symb, vector< string > non_term)

Returns the index in the vector of the symbol with these name.

Definition at line 387 of file Semantics_checks.cpp.

Referenced by genevalmag::Semantics_checks::check_reachability().

Here is the caller graph for this function:



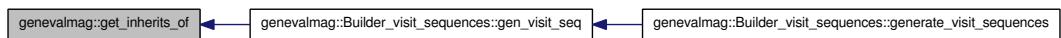
6.1.3.47 void genevalmag::get_inherits_of (const Symbol * symb, const vector< Expr_instance > & computed, vector< Expr_instance > & rec_child)

Obtains the instances of inherit attributes that this symbol.

Definition at line 61 of file Builder_visit_sequences.cpp.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq().

Here is the caller graph for this function:



6.1.3.48 void genevalmag::increment_level (char name)

Definition at line 663 of file Semantics_actions.cpp.

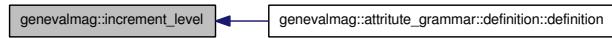
References genevalmag::Semantics_checks::increment_precedence_level(), and sem_check.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.49 void genevalmag::inic_func (const iterator_t str, const iterator_t end)

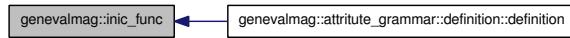
Methods and functions for parse Function.

Definition at line 98 of file Semantics_actions.cpp.

References current_func.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the caller graph for this function:



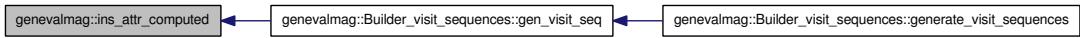
6.1.3.50 bool genevalmag::ins_attr_computed (const Expr_instance * ins, const vector< Expr_instance > & vec)

Searches this instance on the list passed as parameter. If found the instance return true.

Definition at line 46 of file Builder_visit_sequences.cpp.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq().

Here is the caller graph for this function:

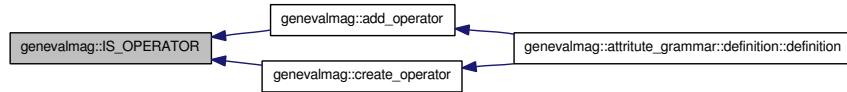


6.1.3.51 const bool genevalmag::IS_OPERATOR (true)

This constant is used for set a function on operator.

Referenced by add_operator(), and create_operator().

Here is the caller graph for this function:



6.1.3.52 const unsigned short genevalmag::LEAVE (0)

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq().

Here is the caller graph for this function:



6.1.3.53 void genevalmag::merge_vec (const vector< map< Key_plan, unsigned short >::const_iterator > & vec_source, vector< map< Key_plan, unsigned short >::const_iterator > & vec_targed)

Merge two vector in the vec_target argument.

Definition at line 93 of file Builder_visit_sequences.cpp.

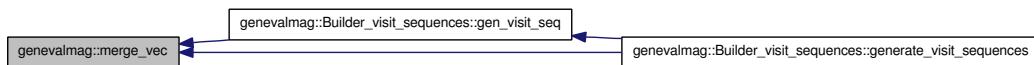
References belong_it().

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq() and genevalmag::Builder_visit_sequences::generate_visit_sequences().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.54 void genevalmag::merge_vec_without_plan (const vector< map< Key_plan, unsigned short >::const_iterator > & vec_source, vector< map< Key_plan, unsigned short >::const_iterator > & vec_targed, const map< Key_plan, unsigned short >::const_iterator & plan)

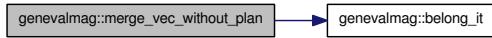
Merge two vector in the vec_target argument without the iterator passed as parameter.

Definition at line 117 of file Builder_visit_sequences.cpp.

References belong_it().

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.55 std::ostream& genevalmag::operator<< (std::ostream & out, file_position const & lc)

Our error reporting parsers

Definition at line 54 of file Parser_AG.cpp.

6.1.3.56 bool genevalmag::parse_parameters (int argc, char * argv[], string & path_input_file, string & path_folder_output, string & name_library, vector< string > & headers)

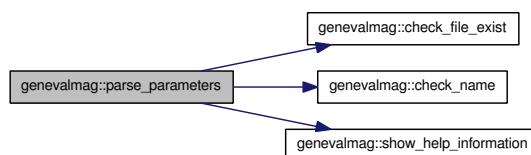
maggen [OPTIONS] where OPTIONS include: -f file Define the input file of the tool. Othercase, uses the standart input (cin). -i header Including header '.h' or '.hpp' file for resolve externs variables in generated code. -fo folder Defines the folder output for output generated information. Othercase, uses "./out_maggen/". -o name Defines the names files generated tool. Otherwise, uses "mag_eval". -h Show this help message.

Definition at line 165 of file maggen.cpp.

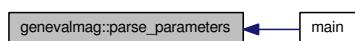
References check_file_exist(), check_name(), and show_help_information().

Referenced by main().

Here is the call graph for this function:



Here is the caller graph for this function:

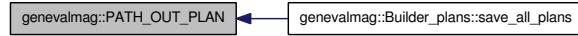


6.1.3.57 const string genevalmag::PATH_OUT_PLAN ("plans/")

Defaults path to saves evaluation plans y their projections.

Referenced by genevalmag::Builder_plans::save_all_plans().

Here is the caller graph for this function:



6.1.3.58 const string genevalmag::PATH_OUT_PLAN_PROJECT ("plans_project/")

Referenced by genevalmag::Builder_plans::save_all_plans_project().

Here is the caller graph for this function:



6.1.3.59 const string genevalmag::PATH_OUTPUT_ADP ("4_AdP_graphs/")

Referenced by genevalmag::Builder_graphs::save_adp_graphs().

Here is the caller graph for this function:



6.1.3.60 const string genevalmag::PATH_OUTPUT_CYCLIC ("CYCLIC_graphs/")

Referenced by genevalmag::Builder_graphs::save_cyclic_graphs().

Here is the caller graph for this function:



6.1.3.61 const string genevalmag::PATH_OUTPUT_DCG ("3_DCG_graphs/")

Referenced by genevalmag::Builder_graphs::save_dcg_graphs().

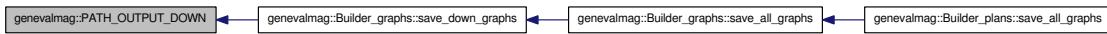
Here is the caller graph for this function:



6.1.3.62 const string genevalmag::PATH_OUTPUT_DOWN ("2_DOWN_graphs/")

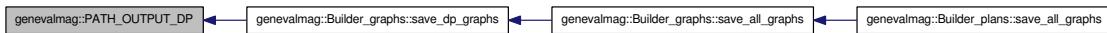
Referenced by genevalmag::Builder_graphs::save_down_graphs().

Here is the caller graph for this function:

**6.1.3.63 const string genevalmag::PATH_OUTPUT_DP ("1_DP_graphs/")**

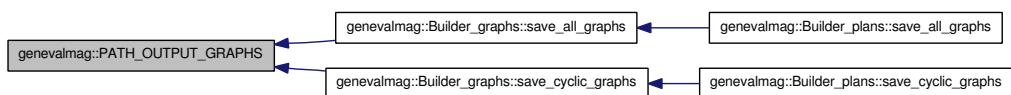
Referenced by genevalmag::Builder_graphs::save_dp_graphs().

Here is the caller graph for this function:

**6.1.3.64 const string genevalmag::PATH_OUTPUT_GRAPHS ("graphs/")**

Referenced by genevalmag::Builder_graphs::save_all_graphs(), and genevalmag::Builder_graphs::save_cyclic_graphs().

Here is the caller graph for this function:

**6.1.3.65 void genevalmag::plan_family_computed (const vector< map< Key_plan, unsigned short >::const_iterator > & plans_computed, vector< unsigned short > & visit_seq_computed)**

Converts the vector of iterators over evaluations plans, in their indexs inside of plans uniques.

Definition at line 144 of file Builder_visit_sequences.cpp.

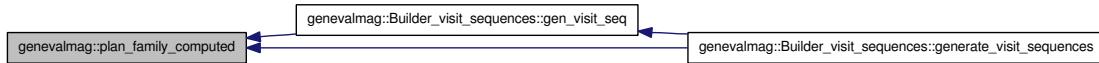
References utilities::belong_index().

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), and genevalmag::Builder_visit_sequences::generate_visit_sequences().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.66 void genevalmag::purge_plan_with (const Rule & rule, const Order_eval_eq & order_eq, Order_eval_eq & purged_order)

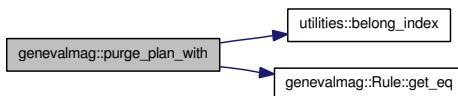
Copy in the result vector all equations belonging to the rule passed as parameter.

Definition at line 321 of file Builder_plans.cpp.

References utilities::belong_index(), and genevalmag::Rule::get_eq().

Referenced by genevalmag::Builder_plans::generate_plans().

Here is the call graph for this function:



Here is the caller graph for this function:



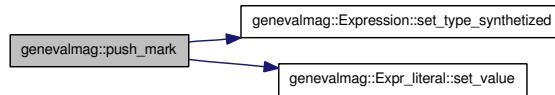
6.1.3.67 void genevalmag::push_mark (char name)

Definition at line 429 of file Semantics_actions.cpp.

References current_literal, genevalmag::Expression::set_type_synthesized(), genevalmag::Expr_literal::set_value(), and stack_node.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



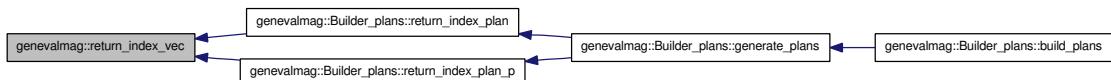
6.1.3.68 `unsigned short genevalmag::return_index_vec (const Order_eval_eq & order, vector< Order_eval_eq > & vec)`

Returns the index in the vector, or inserts in the last position.

Definition at line 370 of file Builder_plans.cpp.

Referenced by genevalmag::Builder_plans::return_index_plan(), and genevalmag::Builder_plans::return_index_plan_p().

Here is the caller graph for this function:



6.1.3.69 `void genevalmag::save_assoc_op (const iterator_t str, const iterator_t end)`

Definition at line 152 of file Semantics_actions.cpp.

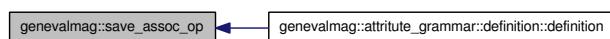
References current_func, and genevalmag::Function::set_oper_assoc().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



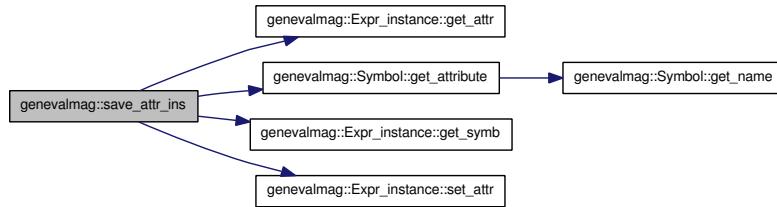
6.1.3.70 `void genevalmag::save_attr_ins (const iterator_t str, const iterator_t end)`

Definition at line 298 of file Semantics_actions.cpp.

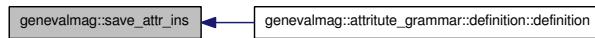
References current_instance, genevalmag::Expr_instance::get_attr(), genevalmag::Symbol::get_attribute(), genevalmag::Expr_instance::get_symb(), and genevalmag::Expr_instance::set_attr().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



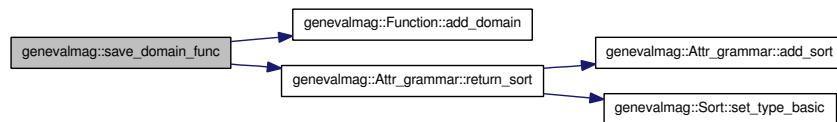
6.1.3.71 void genevalmag::save_domain_func (const iterator_t str, const iterator_t end)

Definition at line 120 of file Semantics_actions.cpp.

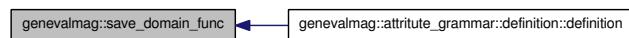
References genevalmag::Function::add_domain(), attr_grammar, current_func, and genevalmag::Attr_grammar::return_sort().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



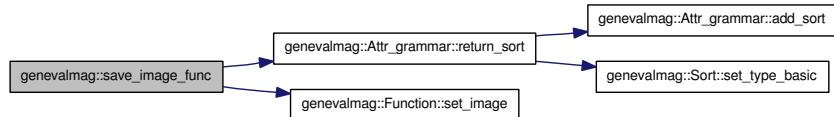
6.1.3.72 void genevalmag::save_image_func (const iterator_t str, const iterator_t end)

Definition at line 126 of file Semantics_actions.cpp.

References attr_grammar, current_func, genevalmag::Attr_grammar::return_sort(), and genevalmag::Function::set_image().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



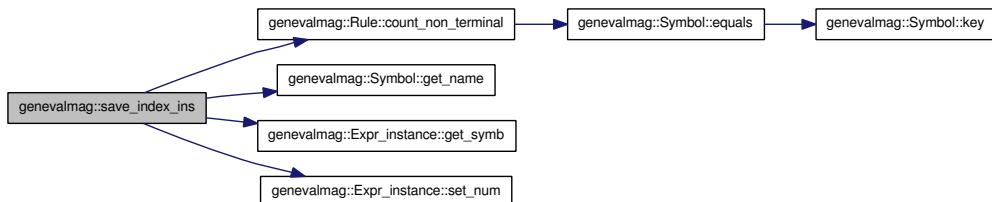
6.1.3.73 void genevalmag::save_index_ins (int const index)

Definition at line 288 of file Semantics_actions.cpp.

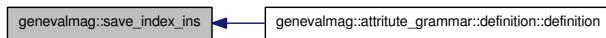
References genevalmag::Rule::count_non_terminal(), current_instance, current_rule, genevalmag::Symbol::get_name(), genevalmag::Expr_instance::get_symb(), and genevalmag::Expr_instance::set_num().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.74 void genevalmag::save_member_list_attr (const iterator_t str, const iterator_t end)

Definition at line 195 of file Semantics_actions.cpp.

References genevalmag::decl_attribute::d_member_symbol, and newAttrs.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the caller graph for this function:



6.1.3.75 void genevalmag::save_mode_op (const iterator_t str, const iterator_t end)

Definition at line 141 of file Semantics_actions.cpp.

References current_func, and genevalmag::Function::set_mode().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.76 void genevalmag::save_name_func (const iterator_t str, const iterator_t end)

Definition at line 114 of file Semantics_actions.cpp.

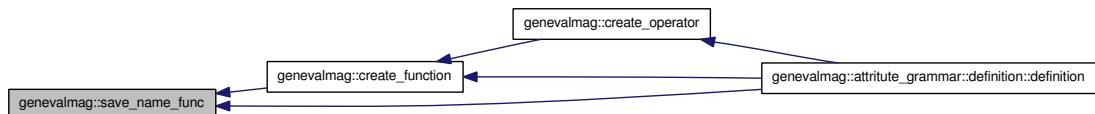
References current_func, and genevalmag::Function::set_name().

Referenced by create_function(), and genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



6.1.3.77 void genevalmag::save_prec_op (int const prec)

Definition at line 147 of file Semantics_actions.cpp.

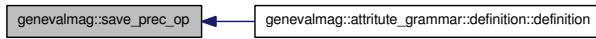
References current_func, and genevalmag::Function::set_prec().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



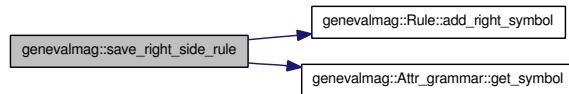
6.1.3.78 void genevalmag::save_right_side_rule (const iterator_t str, const iterator_t end)

Definition at line 247 of file Semantics_actions.cpp.

References genevalmag::Rule::add_right_symbol(), attr_grammar, current_rule, and genevalmag::Attr_grammar::get_symbol().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



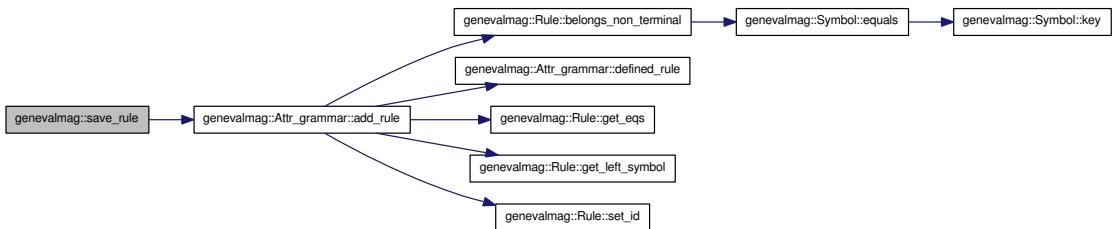
6.1.3.79 void genevalmag::save_rule (const iterator_t str, const iterator_t end)

Definition at line 259 of file Semantics_actions.cpp.

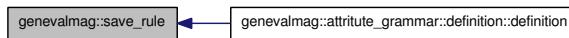
References genevalmag::Attr_grammar::add_rule(), attr_grammar, and current_rule.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



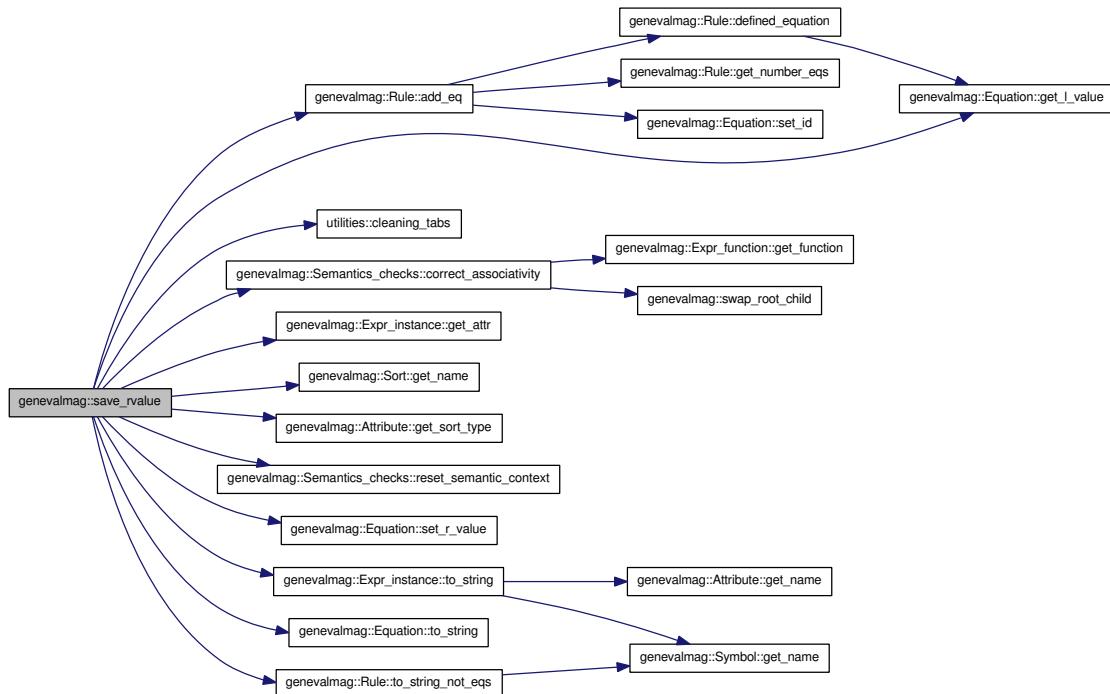
6.1.3.80 void genevalmag::save_rvalue (const iterator_t str, const iterator_t end)

Definition at line 393 of file Semantics_actions.cpp.

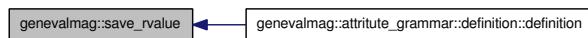
References genevalmag::Rule::add_eq(), utilities::cleaning_tabs(), genevalmag::Semantics_checks::correct_associativity(), current_eq, current_rule, genevalmag::Expr_instance::get_attr(), genevalmag::Equation::get_l_value(), genevalmag::Sort::get_name(), genevalmag::Attribute::get_sort_type(), genevalmag::Semantics_checks::reset_semantic_context(), sem_check, genevalmag::Equation::set_r_value(), stack_node, genevalmag::Expr_instance::to_string(), genevalmag::Equation::to_string(), and genevalmag::Rule::to_string_not_eqs().

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the call graph for this function:



Here is the caller graph for this function:



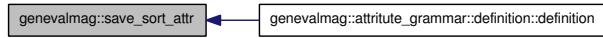
6.1.3.81 void genevalmag::save_sort_attr (const iterator_t str, const iterator_t end)

Definition at line 180 of file Semantics_actions.cpp.

References genevalmag::decl_attribute::d_sort_type, and newAttrs.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the caller graph for this function:



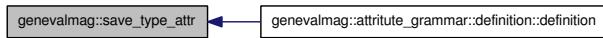
6.1.3.82 void genevalmag::save_type_attr (const iterator_t str, const iterator_t end)

Definition at line 186 of file Semantics_actions.cpp.

References genevalmag::decl_attribute::d_mod_type, k_inherit, and newAttrs.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

Here is the caller graph for this function:



6.1.3.83 void genevalmag::set_attr_grammar (Attr_grammar * at_grammar)

Sets attribute attr_grammar.

Parameters

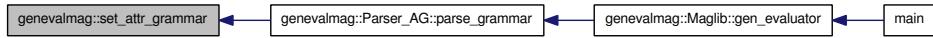
at_grammar Sets attribute attr_grammar.

Definition at line 33 of file Semantics_actions.cpp.

References attr_grammar.

Referenced by genevalmag::Parser_AG::parse_grammar().

Here is the caller graph for this function:



6.1.3.84 void genevalmag::set_sem_check (Semantics_checks * s_check)

Sets attribute sem_check.

Parameters

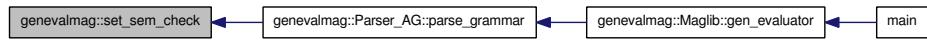
s_check Sets attribute sem_check.

Definition at line 41 of file Semantics_actions.cpp.

References sem_check.

Referenced by genevalmag::Parser_AG::parse_grammar().

Here is the caller graph for this function:

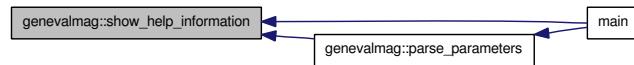


6.1.3.85 void genevalmag::show_help_information ()

Definition at line 144 of file maggen.cpp.

Referenced by main(), and parse_parameters().

Here is the caller graph for this function:



6.1.3.86 int genevalmag::swap_root_child (Expr_function ** old_root, int i_new_root)

Precedence Section. Swaps the root for a left/right child with the correspondent rotations.

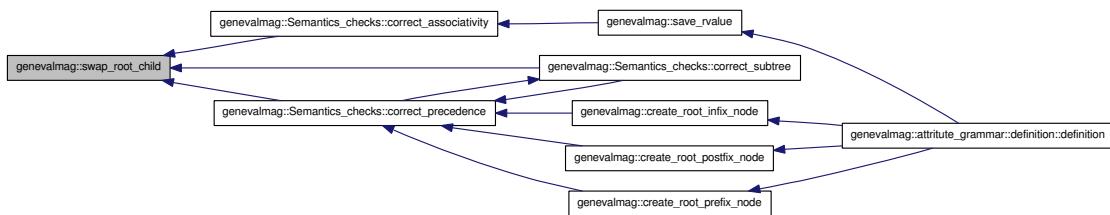
Case 1 A(op) B(op) / \ / \ B C -----> D A / \ / \ D E E C

Case 2 A(op) C(op) / \ / \ B C -----> A E / \ / \ D E B D

Definition at line 107 of file Semantics_checks.cpp.

Referenced by genevalmag::Semantics_checks::correct_associativity(), genevalmag::Semantics_checks::correct_precedence(), and genevalmag::Semantics_checks::correct_subtree().

Here is the caller graph for this function:



6.1.3.87 int genevalmag::swap_root_grandson (Expr_function ** old_root)

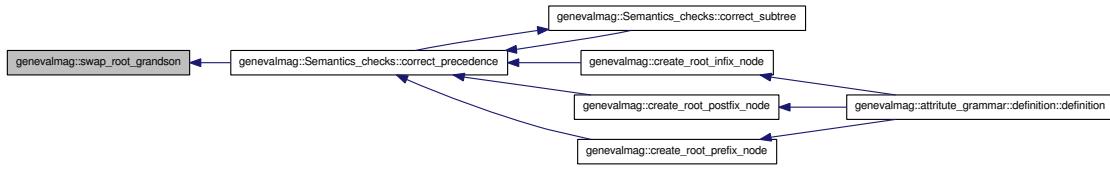
Swaps the root for left-most grandson.

A (op) B(op) / \ / \ B C -----> D E / \ \ D E(op) A \ / \ F F C

Definition at line 145 of file Semantics_checks.cpp.

Referenced by genevalmag::Semantics_checks::correct_precedence().

Here is the caller graph for this function:



6.1.3.88 double genevalmag::timeval_diff (struct timeval * a, struct timeval * b)

Definition at line 109 of file maggen.cpp.

Referenced by main().

Here is the caller graph for this function:



6.1.3.89 template<class K , class T > const string genevalmag::to_string_map (const map< K , T > & map_elem) [inline]

Operation template that print each element type T of the parameter map.

Returns the string representation of all elements of the map.

Definition at line 54 of file Attr_grammar.cpp.

6.1.3.90 void genevalmag::warshall_algorithm (const unsigned int size, bool * matrix_plain)

Computes the closure transitive with the Warshall algorithm.

Definition at line 367 of file Semantics_checks.cpp.

Referenced by genevalmag::Semantics_checks::check_reachability().

Here is the caller graph for this function:



6.1.3.91 template<class T > string genevalmag::write_vector_with_inic (string & text_buffer, const string name_vec, const size_t index, const vector< T > & vec, const string type_vec, const string type_array) [inline]

Generates the initialice of an array of type T, with the elements of the vector, and create a new vector with this array.

Definition at line 227 of file Builder_code.cpp.

6.1.4 Variable Documentation

6.1.4.1 Attr_grammar* genevalmag::attr_grammar

Definition at line 26 of file Semantics_actions.cpp.

Referenced by add_function(), check_well_defined(), create_attributes(), create_instance(), create_new_non_terminal(), create_new_terminal(), create_root_function_node(), create_root_infix_node(), create_root_postfix_node(), create_root_prefix_node(), create_rule(), create_sort(), genevalmag::Builder_visit_sequences::gen_visit_seq(), save_domain_func(), save_image_func(), save_right_side_rule(), save_rule(), and set_attr_grammar().

6.1.4.2 Expr_function* genevalmag::current_ast_function

Definition at line 72 of file Semantics_actions.cpp.

Referenced by create_func_node().

6.1.4.3 Equation* genevalmag::current_eq

Definition at line 73 of file Semantics_actions.cpp.

Referenced by create_equation(), and save_rvalue().

6.1.4.4 Function* genevalmag::current_func

Pointer that reference a new function in the grammar.

Definition at line 49 of file Semantics_actions.cpp.

Referenced by add_function(), add_operator(), create_func_node(), create_function(), create_operator(), inic_func(), save_assoc_op(), save_domain_func(), save_image_func(), save_mode_op(), save_name_func(), and save_prec_op().

6.1.4.5 Expr_instance* genevalmag::current_instance

Pointer to the last instance of attribute to parse successfully.

Definition at line 70 of file Semantics_actions.cpp.

Referenced by create_equation(), create_instance(), create_instance_node(), save_attr_ins(), and save_index_ins().

6.1.4.6 Expr_literal* genevalmag::current_literal

Definition at line 71 of file Semantics_actions.cpp.

Referenced by create_bool(), create_lit_ch(), create_lit_number(), create_lit_str(), create_literal_node(), and push_mark().

6.1.4.7 Rule* genevalmag::current_rule

Pointer that reference a current rule parsed in the grammar.

Definition at line 65 of file Semantics_actions.cpp.

Referenced by genevalmag::Builder_graphs::compute_adp_graph(), genevalmag::Builder_graphs::compute_dcg(), genevalmag::Builder_graphs::compute_down_graph(), create_abbreviated_rule(), create_instance(), create_rule(), genevalmag::Builder_graphs::save_dcg_graphs(), genevalmag::Builder_graphs::save_dp_graphs(), save_index_ins(), save_right_side_rule(), save_rule(), and save_rvalue().

6.1.4.8 `vector<Equation*> genevalmag::index_access_eq`

Definition at line 20 of file Rule.cpp.

6.1.4.9 `struct genevalmag::decl_attribute * genevalmag::new_attrs`

Type that represent the structure of a full declaration of one [Attribute](#).

Referenced by add_attribute(), create_attributes(), save_member_list_attr(), save_sort_attr(), and save_type_attr().

6.1.4.10 `Semantics_checks* genevalmag::sem_check`

Definition at line 28 of file Semantics_actions.cpp.

Referenced by check_well_defined(), create_func_node(), create_root_infix_node(), create_root_postfix_node(), create_root_prefix_node(), decrement_level(), increment_level(), save_rvalue(), and set_sem_check().

6.1.4.11 `vector<Expr_node*> genevalmag::stack_inner_node`

Definition at line 79 of file Semantics_actions.cpp.

Referenced by create_func_node(), create_root_function_node(), create_root_infix_node(), create_root_postfix_node(), and create_root_prefix_node().

6.1.4.12 `vector<Expression*> genevalmag::stack_node`

Stacks for expression precedence manager.

Definition at line 78 of file Semantics_actions.cpp.

Referenced by create_instance_node(), create_literal_node(), create_root_function_node(), create_root_infix_node(), create_root_postfix_node(), create_root_prefix_node(), push_mark(), and save_rvalue().

6.2 utilities Namespace Reference

Functions

- bool `create_folder` (const string path)
- bool `clean_output_folder` (const string path)
- bool `copy_static_code` (const string path_d, const string path_s)
- void `generate_names_instance` (const Graph &graph, string datas[], size_t size_d)
- void `generate_names_attr` (const Graph &graph, string datas[], size_t size_d)
- void `print_graph` (const Graph &graph, const string path, const string name_file, const string name_graph, const string names[], string shape_vertex)
- Vertex `return_vertex` (const Graph &graph, const Expr_leaf *node)
- void `merge_graph` (const Graph &graph1, const Graph &graph2, Graph &graph_merged)
- void `project_graph` (const Symbol *symb, Graph &graph)
- string `cleaning_tabs` (const string str)
- string `write_inf_context` (const vector< unsigned short > &context_vec)
- bool `belong_index` (const unsigned short &index, const vector< unsigned short > &vec)

6.2.1 Function Documentation

6.2.1.1 bool utilities::belong_index (const unsigned short & index, const vector< unsigned short > & vec)

Searches a index in a vector.

Parameters

index

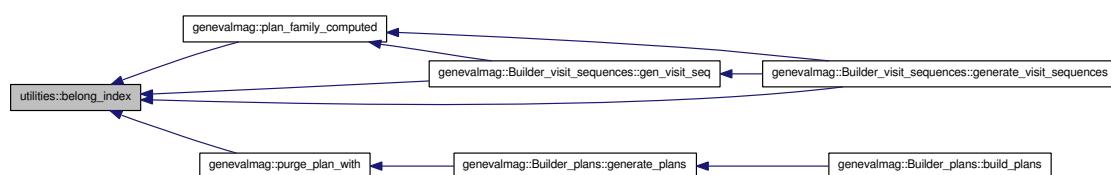
vec

Returns

Definition at line 211 of file Utilities.cpp.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), genevalmag::Builder_visit_sequences::generate_visit_sequences(), genevalmag::plan_family_computed(), and genevalmag::purge_plan_with().

Here is the caller graph for this function:



6.2.1.2 bool utilities::clean_output_folder (const string path)

Remove and create the output folder of files .dot and .png.

Parameters

path

Returns

Remove and create the output folder of files .dot and .png.

Definition at line 41 of file Utilities.cpp.

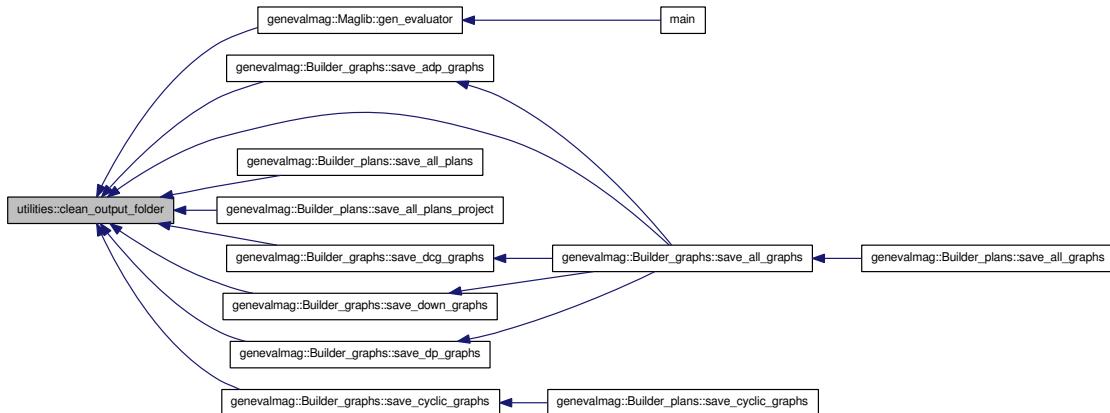
References create_folder().

Referenced by genevalmag::Maglib::gen_evaluator(), genevalmag::Builder_graphs::save_adp_graphs(), genevalmag::Builder_graphs::save_all_graphs(), genevalmag::Builder_plans::save_all_plans(), genevalmag::Builder_plans::save_all_plans_project(), genevalmag::Builder_graphs::save_cyclic_graphs(), genevalmag::Builder_graphs::save_dcg_graphs(), genevalmag::Builder_graphs::save_down_graphs(), and genevalmag::Builder_graphs::save_dp_graphs().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.1.3 string utilities::cleaning_tabs (const string str)

Remove tabs and replace for spaces.

Parameters

str

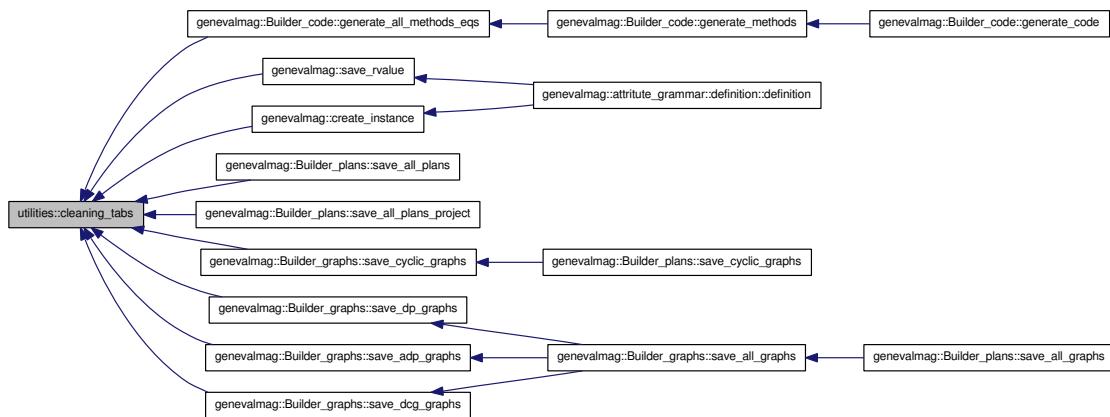
Returns

Remove tabs and replace for spaces.

Definition at line 226 of file Utilities.cpp.

Referenced by genevalmag::create_instance(), genevalmag::Builder_code::generate_all_methods_eqs(), genevalmag::Builder_graphs::save_adp_graphs(), genevalmag::Builder_plans::save_all_plans(), genevalmag::Builder_plans::save_all_plans_project(), genevalmag::Builder_graphs::save_cyclic_graphs(), genevalmag::Builder_graphs::save_dcg_graphs(), genevalmag::Builder_graphs::save_dp_graphs(), and genevalmag::save_rvalue().

Here is the caller graph for this function:



6.2.1.4 bool utilities::copy_static_code (const string path_d, const string path_s)

Copies the static file to the generation path.

Parameters

path_d
path_s

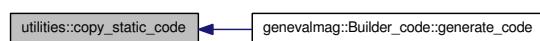
Returns

Copies the static file to the generation path.

Definition at line 57 of file Utilities.cpp.

Referenced by genevalmag::Builder_code::generate_code().

Here is the caller graph for this function:



6.2.1.5 bool utilities::create_folder (const string path)

Create the folder passed as parameter.

Parameters

path

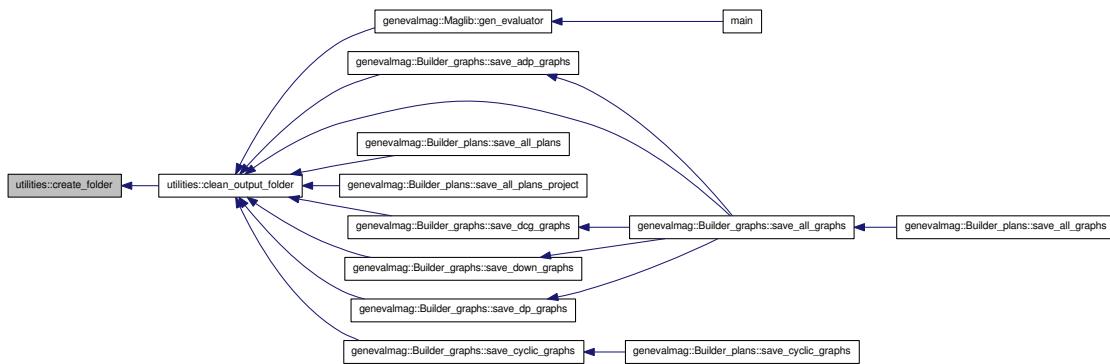
Returns

Create the folder passed as parameter.

Definition at line 26 of file Utilities.cpp.

Referenced by clean_output_folder().

Here is the caller graph for this function:



6.2.1.6 void utilities::generate_names_attr (const Graph & graph, string datas[], size_t size_d)

Generates the names of vertex for down graph. (The vertex's name is an attribute).

Parameters

graph

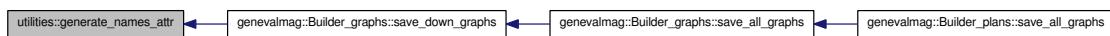
datas

size_d Generates the names of vertex for down graph. (The vertex's name is an attribute).

Definition at line 88 of file Utilities.cpp.

Referenced by genevalmag::Builder_graphs::save_down_graphs().

Here is the caller graph for this function:



6.2.1.7 void utilities::generate_names_instance (const Graph & *graph*, string *datas*[], size_t *size_d*)

Generates the names of vertex. (The vertex's name is an instance).

Parameters

graph

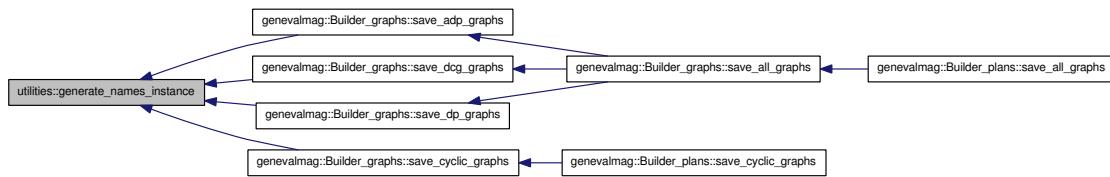
datas

size_d Generates the names of vertex. (The vertex's name is an instance).

Definition at line 75 of file Utilities.cpp.

Referenced by genevalmag::Builder_graphs::save_adp_graphs(), genevalmag::Builder_graphs::save_cyclic_graphs(), genevalmag::Builder_graphs::save_dcg_graphs(), and genevalmag::Builder_graphs::save_dp_graphs().

Here is the caller graph for this function:



6.2.1.8 void utilities::merge_graph (const Graph & *graph1*, const Graph & *graph2*, Graph & *graph_merged*)

Joins graph1 and graph2 in graph_merged.

Parameters

graph1

graph2

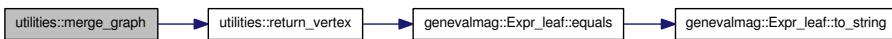
graph_merged Joins graph1 and graph2 in graph_merged.

Definition at line 152 of file Utilities.cpp.

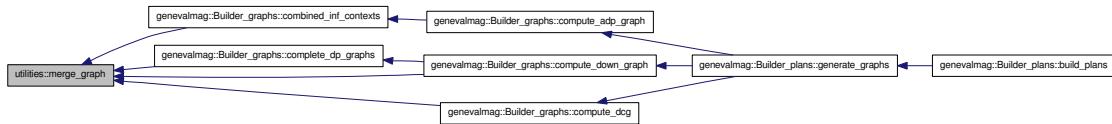
References return_vertex().

Referenced by genevalmag::Builder_graphs::combined_inf_contexts(), genevalmag::Builder_graphs::complete_dp_graphs(), genevalmag::Builder_graphs::compute_dcg(), and genevalmag::Builder_graphs::compute_down_graph().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.1.9 void utilities::print_graph (const Graph & graph, const string path, const string name_file, const string name_graph, const string names[], string shape_vertex)

Prints a graph in a file .dot for generate image .spng.

Parameters

graph

path

name_file

name_graph

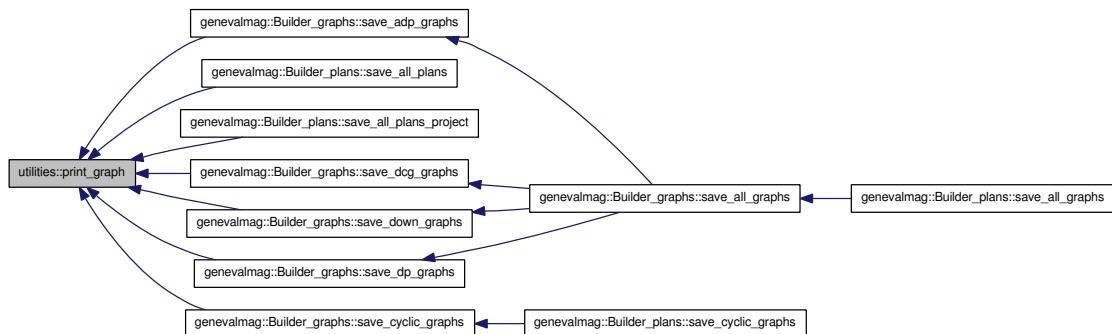
names

shape_vertex Prints a graph in a file .dot for generate image .spng.

Definition at line 102 of file Utilities.cpp.

Referenced by genevalmag::Builder_graphs::save_adp_graphs(), genevalmag::Builder_plans::save_all_plans(), genevalmag::Builder_plans::save_all_plans_project(), genevalmag::Builder_graphs::save_cyclic_graphs(), genevalmag::Builder_graphs::save_dcg_graphs(), genevalmag::Builder_graphs::save_down_graphs(), and genevalmag::Builder_graphs::save_dp_graphs().

Here is the caller graph for this function:



6.2.1.10 void utilities::project_graph (const Symbol * symb, Graph & graph)

Projects a graph with only vertex that belongs to symbol "symb". Modifies the parameter "graph".

Parameters***symb***

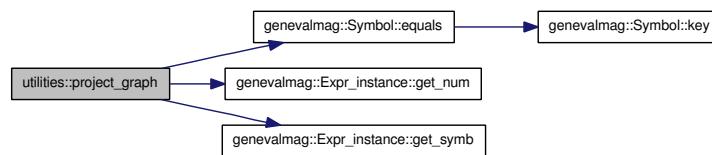
graph Projects a graph with only vertex that belongs to symbol "symb". Modifies the parameter "graph".

Definition at line 192 of file Utilities.cpp.

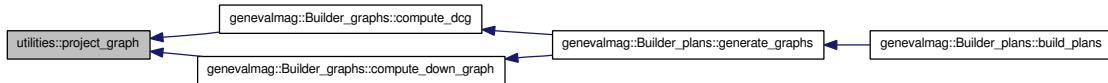
References genevalmag::Symbol::equals(), genevalmag::Expr_instance::get_num(), and genevalmag::Expr_instance::get_symb().

Referenced by genevalmag::Builder_graphs::compute_dcg(), and genevalmag::Builder_graphs::compute_down_graph().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.1.11 Vertex utilities::return_vertex (const Graph & *graph*, const Expr_leaf * *node*)

Given a graph and node, returns the vertex descriptor of node in the graph. If not search it, so returns USHRT_MAX.

Parameters***graph******node*****Returns**

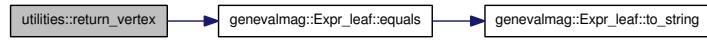
Given a graph and node, returns the vertex descriptor of node in the graph. If not search it, so returns USHRT_MAX.

Definition at line 138 of file Utilities.cpp.

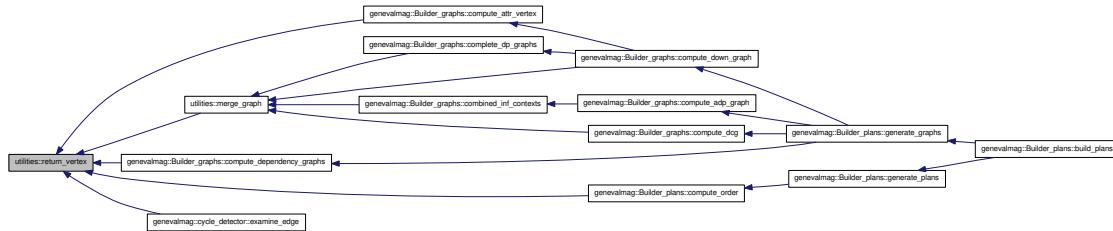
References genevalmag::Expr_leaf::equals().

Referenced by genevalmag::Builder_graphs::compute_attr_vertex(), genevalmag::Builder_graphs::compute_dependency_graphs(), genevalmag::Builder_plans::compute_order(), genevalmag::cycle_detector::examine_edge(), and merge_graph().

Here is the call graph for this function:



Here is the caller graph for this function:



6.2.1.12 string utilities::write_inf_context (const vector< unsigned short > & context_vec)

Writes inferior contex in a string and returns it.

Parameters

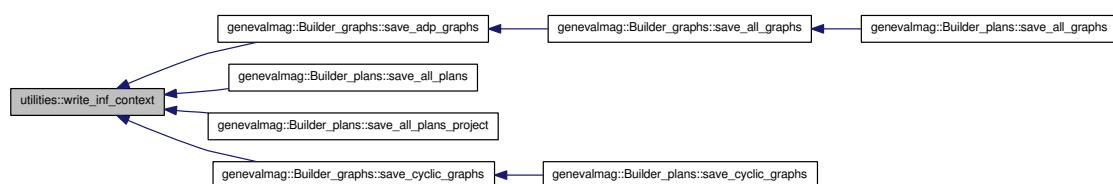
context_vec

Returns

Definition at line 236 of file Utilities.cpp.

Referenced by genevalmag::Builder_graphs::save_adp_graphs(), genevalmag::Builder_plans::save_all_plans(), genevalmag::Builder_plans::save_all_plans_project(), and genevalmag::Builder_graphs::save_cyclic_graphs().

Here is the caller graph for this function:



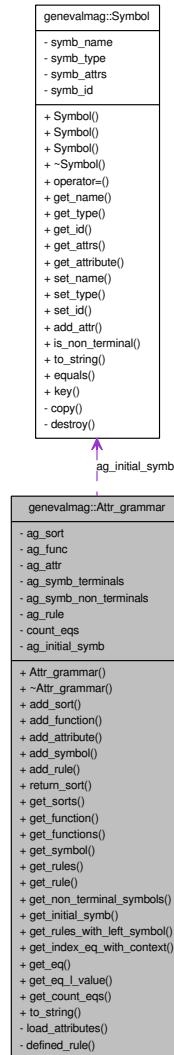
Chapter 7

Class Documentation

7.1 genevalmag::Attr_grammar Class Reference

```
#include <Attr_grammar.h>
```

Collaboration diagram for genevalmag::Attr_grammar:



Public Member Functions

- `Attr_grammar()`
- `virtual ~Attr_grammar()`
- `const bool add_sort(const Sort &sort)`
- `const bool add_function(const Function &func)`
- `const bool add_attribute(const Attribute &attr)`
- `const bool add_symbol(const Symbol &symb)`
- `const bool add_rule(Rule &rule)`
- `const Sort & return_sort(const string name_sort)`
- `const map< string, Sort > & get_sorts() const`
- `const Function * get_function(const string key_function) const`
- `const map< string, Function > & get_functions() const`
- `const Symbol & get_symbol(const string name_symbol) const`

- const map< unsigned short, Rule > & get_rules () const
- const Rule & get_rule (const unsigned short index) const
- const map< string, Symbol > & get_non_terminal_symbols () const
- const Symbol * get_initial_symb () const
- const vector< unsigned short > get_rules_with_left_symbol (const Symbol *symb) const
- const unsigned short get_index_eq_with_context (const Expr_instance *ins, const vector< unsigned short > &context_rule) const
- const Equation * get_eq (const unsigned short index) const
- const Expr_instance * get_eq_1_value (const unsigned short index) const
- const unsigned short get_count_eqs () const
- const string to_string () const

Private Member Functions

- void load_attributes (Symbol &symb) const
- const bool defined_rule (const Rule &rule) const

Private Attributes

- map< string, Sort > ag_sort
- map< string, Function > ag_func
- map< string, Attribute > ag_attr
- map< string, Symbol > ag_symb_terminals
- map< string, Symbol > ag_symb_non_terminals
- map< unsigned short, Rule > ag_rule
- unsigned short count_eqs

Store the count of equations in the grammar.

- const Symbol * ag_initial_symb

Saves the name of the initial symbol of the grammar's attribute.

7.1.1 Detailed Description

Definition at line 25 of file Attr_grammar.h.

7.1.2 Constructor & Destructor Documentation

7.1.2.1 genevalmag::Attr_grammar::Attr_grammar()

Constructor empty of semantic domain.

Constructor empty of attribute grammar.

Initialice values.

Definition at line 22 of file Attr_grammar.cpp.

References ag_initial_symb, and count_eqs.

7.1.2.2 `genevalmag::Attr_grammar::~Attr_grammar () [virtual]`

Destructor of the semantic domain.

Destructor of the attribute grammar.

Definition at line 32 of file Attr_grammar.cpp.

7.1.3 Member Function Documentation

7.1.3.1 `const bool genevalmag::Attr_grammar::add_attribute (const Attribute & attr)`

Enqueues a attribute in the list of the semantic domain.

Parameters

attr

Returns

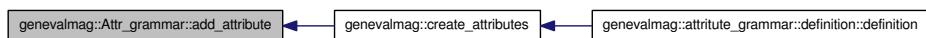
Enqueues a attribute in the list of the attribute grammar.

Definition at line 85 of file Attr_grammar.cpp.

References ag_attr.

Referenced by genevalmag::create_attributes().

Here is the caller graph for this function:



7.1.3.2 `const bool genevalmag::Attr_grammar::add_function (const Function & func)`

Enqueues a function in the list of the semantic domain.

Parameters

func

Returns

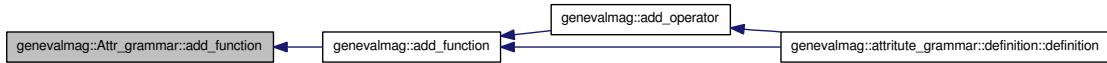
Enqueues a function in the list of the attribute grammar.

Definition at line 77 of file Attr_grammar.cpp.

References ag_func.

Referenced by genevalmag::add_function().

Here is the caller graph for this function:



7.1.3.3 const bool genevalmag::Attr_grammar::add_rule (Rule & rule)

Enqueues a rule in the list of the semantic domain.

Parameters

rule

Returns

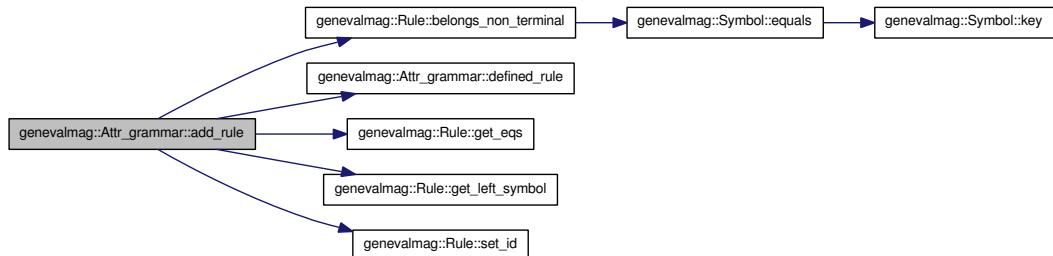
Enqueues a rule in the list of the attribute grammar.

Definition at line 183 of file Attr_grammar.cpp.

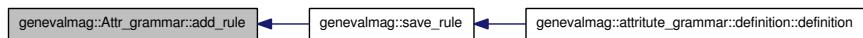
References ag_initial_symb, ag_rule, genevalmag::Rule::belongs_non_terminal(), count_eqs, defined_rule(), genevalmag::Rule::get_eqs(), genevalmag::Rule::get_left_symbol(), and genevalmag::Rule::set_id().

Referenced by genevalmag::save_rule().

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.4 const bool genevalmag::Attr_grammar::add_sort (const Sort & sort)

Enqueues a sort in the list of the semantic domain.

Parameters

sort

Returns

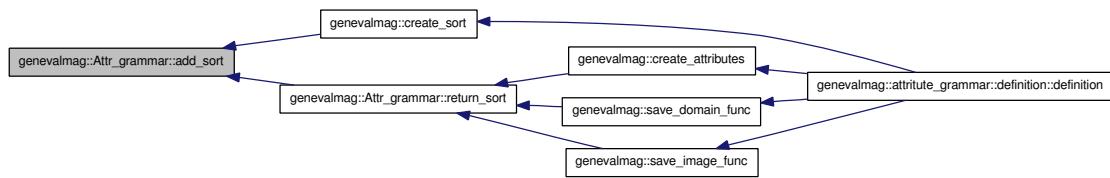
Enqueues a sort in the list of the attribute grammar.

Definition at line 69 of file Attr_grammar.cpp.

References ag_sort.

Referenced by genevalmag::create_sort(), and return_sort().

Here is the caller graph for this function:



7.1.3.5 const bool genevalmag::Attr_grammar::add_symbol (const Symbol & symb)

Enqueues a symbol in the list of the semantic domain.

Parameters

symb

Returns

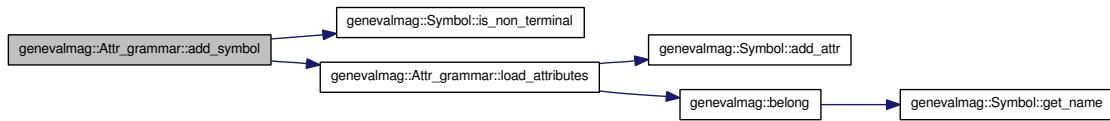
Enqueues a symbol in the list of the attribute grammar.

Definition at line 142 of file Attr_grammar.cpp.

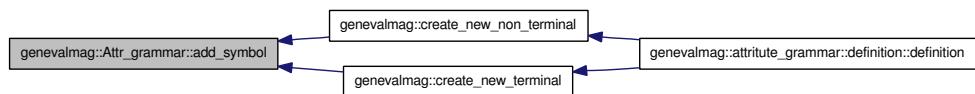
References ag_symb_non_terminals, ag_symb_terminals, genevalmag::Symbol::is_non_terminal(), and load_attributes().

Referenced by genevalmag::create_new_non_terminal(), and genevalmag::create_new_terminal().

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.6 const bool genevalmag::Attr_grammar::defined_rule (const Rule & rule) const [private]

Checks that the rule is not already defined in the grammar.

Parameters

rule

Returns

Checks that the rule is not already defined in the grammar.

Definition at line 168 of file Attr_grammar.cpp.

References ag_rule.

Referenced by add_rule().

Here is the caller graph for this function:



7.1.3.7 const unsigned short genevalmag::Attr_grammar::get_count_eqs () const

Returns the count of equations in the grammar.

Returns

Returns the count of equations in the grammar.

Definition at line 397 of file Attr_grammar.cpp.

References count_eqs.

Referenced by genevalmag::Builder_code::generate_all_compute_eq().

Here is the caller graph for this function:



7.1.3.8 const Equation * genevalmag::Attr_grammar::get_eq (const unsigned short index) const

Returns the equation with this index.

Parameters

index

Returns

Returns the equation with this index.

Definition at line 367 of file Attr_grammar.cpp.

References ag_rule, and count_eqs.

Referenced by get_eq_l_value(), genevalmag::Builder_plans::save_all_plans(), and genevalmag::Builder_plans::save_all_plans_project().

Here is the caller graph for this function:



7.1.3.9 const Expr_instance * genevalmag::Attr_grammar::get_eq_l_value (const unsigned short index) const

Returns the l_value of the equation with this index.

Parameters

index

Returns

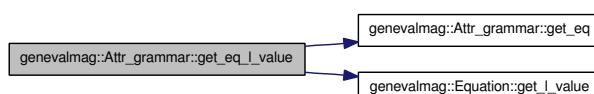
Returns the l_value of the equation with this index.

Definition at line 387 of file Attr_grammar.cpp.

References get_eq(), and genevalmag::Equation::get_l_value().

Referenced by genevalmag::Builder_plans::compute_order().

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.10 **const Function * genevalmag::Attr_grammar::get_function (const string key_function) const**

Finds in the list of function of the semantic domain and returns the function with that name.

Parameters

key_function

Returns

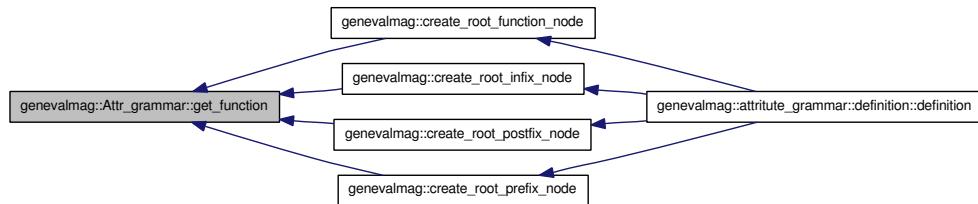
Finds in the list of function of the attribute grammar and returns the function with that name.

Definition at line 246 of file Attr_grammar.cpp.

References ag_func.

Referenced by genevalmag::create_root_function_node(), genevalmag::create_root_infix_node(), genevalmag::create_root_postfix_node(), and genevalmag::create_root_prefix_node().

Here is the caller graph for this function:



7.1.3.11 **const map< string, Function > & genevalmag::Attr_grammar::get_functions () const**

Returns the map with all functions.

Returns

Returns the map with all functions.

Definition at line 259 of file Attr_grammar.cpp.

References ag_func.

7.1.3.12 **const unsigned short genevalmag::Attr_grammar::get_index_eq_with_context (const Expr_instance * ins, const vector< unsigned short > & context_rule) const**

Returns the index of an equation in this range of rules with l_value equals to ins.

Parameters

ins
context_rule

Returns

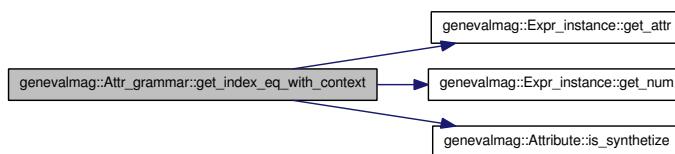
Returns the index of an equation in this range of rules with l_value equals to ins.

Definition at line 329 of file Attr_grammar.cpp.

References ag_rule, genevalmag::Expr_instance::get_attr(), genevalmag::Expr_instance::get_num(), and genevalmag::Attribute::is_synthetize().

Referenced by genevalmag::Builder_plans::generates_topological_order().

Here is the call graph for this function:



Here is the caller graph for this function:

**7.1.3.13 const Symbol * genevalmag::Attr_grammar::get_initial_symb () const**

Returns the initial rule.

Returns

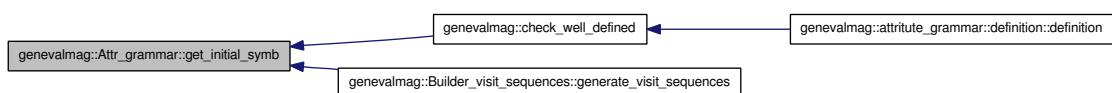
Returns the initial symbol.

Definition at line 304 of file Attr_grammar.cpp.

References ag_initial_symb.

Referenced by genevalmag::check_well_defined(), and genevalmag::Builder_visit_sequences::generate_visit_sequences().

Here is the caller graph for this function:



7.1.3.14 `const map< string, Symbol > & genevalmag::Attr_grammar::get_non_-terminal_symbols () const`

Returns the map with all symbols.

Returns

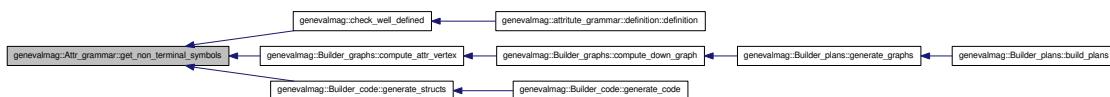
Returns the map with all symbols.

Definition at line 296 of file Attr_grammar.cpp.

References ag_symb_non_terminals.

Referenced by genevalmag::check_well_defined(), genevalmag::Builder_graphs::compute_attr_vertex(), and genevalmag::Builder_code::generate_structs().

Here is the caller graph for this function:



7.1.3.15 `const Rule & genevalmag::Attr_grammar::get_rule (const unsigned short index) const`

Returns the rule on parameterter.

Parameters

index

Returns

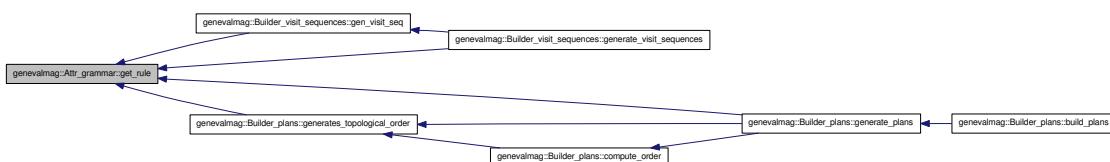
Returns the rule on parameterter.

Definition at line 288 of file Attr_grammar.cpp.

References ag_rule.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), genevalmag::Builder_plans::generate_plans(), genevalmag::Builder_visit_sequences::generate_visit_sequences(), and genevalmag::Builder_plans::generates_topological_order().

Here is the caller graph for this function:



7.1.3.16 `const map< unsigned short, Rule > & genevalmag::Attr_grammar::get_rules()` const

Returns the map with all rules.

Returns

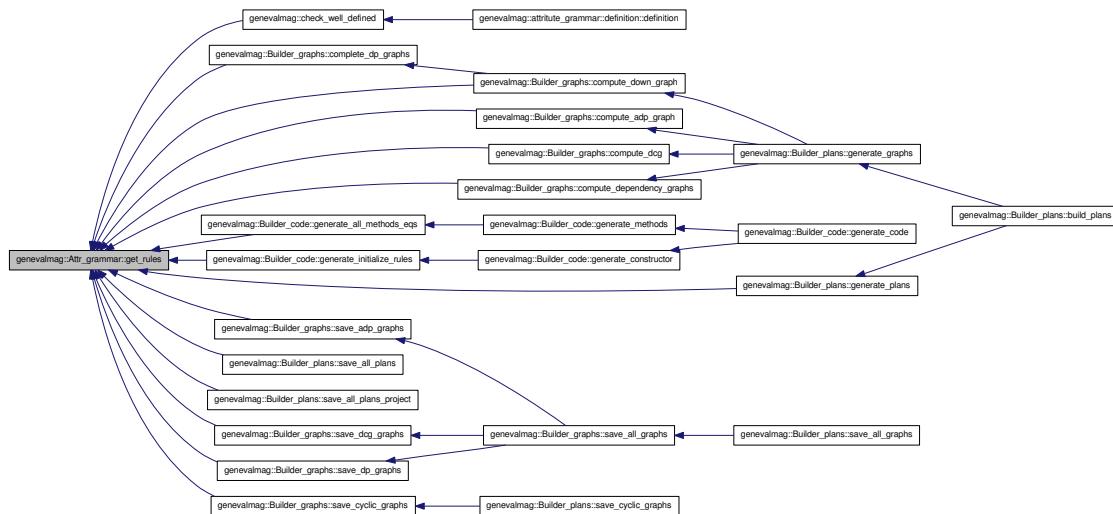
Returns the map with all rules.

Definition at line 280 of file Attr_grammar.cpp.

References ag_rule.

Referenced by genevalmag::check_well_defined(), genevalmag::Builder_graphs::complete_dp_graphs(), genevalmag::Builder_graphs::compute_adp_graph(), genevalmag::Builder_graphs::compute_dcg(), genevalmag::Builder_graphs::compute_dependency_graphs(), genevalmag::Builder_graphs::compute_down_graph(), genevalmag::Builder_code::generate_all_methods_eqs(), genevalmag::Builder_code::generate_initialize_rules(), genevalmag::Builder_plans::generate_plans(), genevalmag::Builder_graphs::save_adp_graphs(), genevalmag::Builder_plans::save_all_plans(), genevalmag::Builder_plans::save_all_plans_project(), genevalmag::Builder_graphs::save_cyclic_graphs(), genevalmag::Builder_graphs::save_dcg_graphs(), and genevalmag::Builder_graphs::save_dp_graphs().

Here is the caller graph for this function:



7.1.3.17 `const vector< unsigned short > genevalmag::Attr_grammar::get_rules_with_left_symbol(const Symbol * symb)` const

Returns vector with all rules with the left symbol equal that parameter.

Parameters

symb

Returns

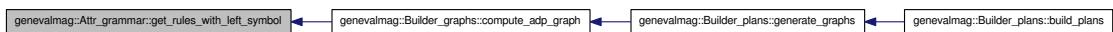
Returns vector with all rules with the left symbol equal that parameter.

Definition at line 312 of file Attr_grammar.cpp.

References ag_rule.

Referenced by genevalmag::Builder_graphs::compute_adp_graph().

Here is the caller graph for this function:

**7.1.3.18 const map< string, Sort > & genevalmag::Attr_grammar::get_sorts () const**

Returns the map with all sorts.

Returns

Returns the map with all sorts.

Definition at line 238 of file Attr_grammar.cpp.

References ag_sort.

7.1.3.19 const Symbol & genevalmag::Attr_grammar::get_symbol (const string name_symbol) const

Finds in the list of symbol of the semantic domain and returns the symbol with that name.

Parameters

name_symbol

Returns

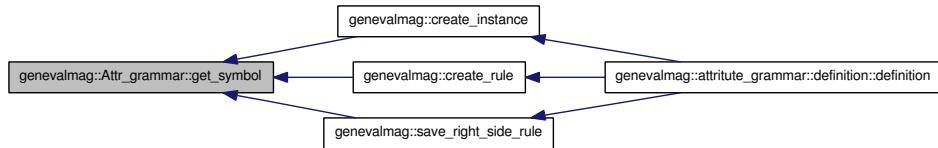
Finds in the list of operator of the attribute grammar and returns the operator with that name.

Definition at line 267 of file Attr_grammar.cpp.

References ag_symb_non_terminals, and ag_symb_terminals.

Referenced by genevalmag::create_instance(), genevalmag::create_rule(), and genevalmag::save_right_side_rule().

Here is the caller graph for this function:



7.1.3.20 void genevalmag::Attr_grammar::load_attributes (Symbol & symb) const [private]

Insert the attributes belong the symbol.

Parameters

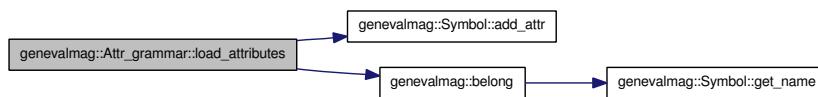
symb Inserts the attributes belong the symbol.

Definition at line 128 of file Attr_grammar.cpp.

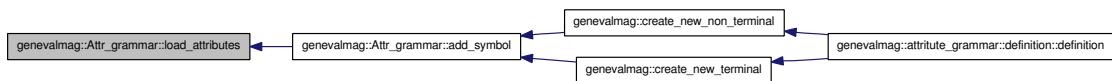
References genevalmag::Symbol::add_attr(), ag_attr, and genevalmag::belong().

Referenced by add_symbol().

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.21 const Sort & genevalmag::Attr_grammar::return_sort (const string name_sort)

Finds in the list of sort of the semantic domain and returns the sort with that name.

Parameters

name_sort

Returns

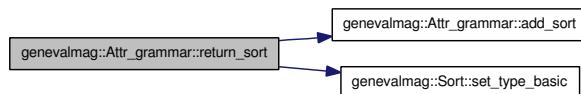
Finds in the list of sort of the attribute grammar and returns the sort with that name.

Definition at line 225 of file Attr_grammar.cpp.

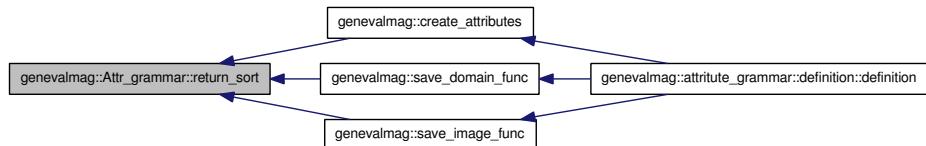
References add_sort(), ag_sort, and genevalmag::Sort::set_type_basic().

Referenced by genevalmag::create_attributes(), genevalmag::save_domain_func(), and genevalmag::save_image_func().

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.3.22 const string genevalmag::Attr_grammar::to_string () const

Generates and returns a string representation of a semantic domain.

Result = "semantic domain"

```

<sorts>
<operators>
<functions>
"attributes"
<attributes>

```

[This section is commented so that it can be parsed again.]

```
*****
```

```

"symbols"
<symbols>
*****
```

```
"rules"
```

```
<rules>
```

where <sorts>, <operators>, <functions>, <attributes>, <symbols> and <rules>, are a full representation of each type.

Returns

Generates and returns a string representation of a attribute grammar. where sorts, operators, functions, attributes, symbols and rules, are full representation of each type.

Definition at line 407 of file Attr_grammar.cpp.

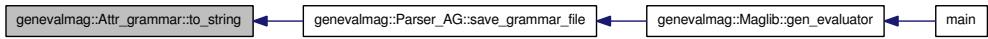
References ag_attr, ag_func, ag_initial_symb, ag_rule, ag_sort, ag_symb_non_terminals, ag_symb_terminals, and genevalmag::Symbol::get_name().

Referenced by genevalmag::Parser_AG::save_grammar_file().

Here is the call graph for this function:



Here is the caller graph for this function:



7.1.4 Member Data Documentation

7.1.4.1 map<string, Attribute> genevalmag::Attr_grammar::ag_attr [private]

Definition at line 39 of file Attr_grammar.h.

Referenced by add_attribute(), load_attributes(), and to_string().

7.1.4.2 map<string, Function> genevalmag::Attr_grammar::ag_func [private]

Definition at line 38 of file Attr_grammar.h.

Referenced by add_function(), get_function(), get_functions(), and to_string().

7.1.4.3 genevalmag::Attr_grammar::ag_initial_symb [private]

Saves the name of the initial symbol of the grammar's attribute.

Definition at line 54 of file Attr_grammar.h.

Referenced by add_rule(), Attr_grammar(), get_initial_symb(), and to_string().

7.1.4.4 map<unsigned short, Rule> genevalmag::Attr_grammar::ag_rule [private]

Definition at line 42 of file Attr_grammar.h.

Referenced by add_rule(), defined_rule(), get_eq(), get_index_eq_with_context(), get_rule(), get_rules(), get_rules_with_left_symbol(), and to_string().

7.1.4.5 map<string, Sort> genevalmag::Attr_grammar::ag_sort [private]

Definition at line 37 of file Attr_grammar.h.

Referenced by add_sort(), get_sorts(), return_sort(), and to_string().

7.1.4.6 `map<string, Symbol> genevalmag::Attr_grammar::ag_symb_non_terminals [private]`

Definition at line 41 of file Attr_grammar.h.

Referenced by add_symbol(), get_non_terminal_symbols(), get_symbol(), and to_string().

7.1.4.7 `map<string, Symbol> genevalmag::Attr_grammar::ag_symb_terminals [private]`

Definition at line 40 of file Attr_grammar.h.

Referenced by add_symbol(), get_symbol(), and to_string().

7.1.4.8 `genevalmag::Attr_grammar::count_eqs [private]`

Store the count of equations in the grammar.

Definition at line 48 of file Attr_grammar.h.

Referenced by add_rule(), Attr_grammar(), get_count_eqs(), and get_eq().

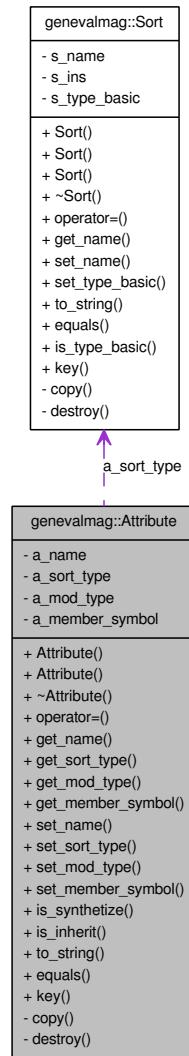
The documentation for this class was generated from the following files:

- include/Attr_grammar/Attr_grammar.h
- src/Attr_grammar/Attr_grammar.cpp

7.2 genevalmag::Attribute Class Reference

```
#include <Attribute.h>
```

Collaboration diagram for genevalmag::Attribute:



Public Member Functions

- [Attribute \(\)](#)
- [Attribute \(const Attribute &other\)](#)
- [virtual ~Attribute \(\)](#)
- [Attribute & operator= \(const Attribute &other\)](#)
- [const string get_name \(\) const](#)
- [const Sort * get_sort_type \(\) const](#)
- [const type_attr get_mod_type \(\) const](#)
- [const string get_member_symbol \(\) const](#)
- [void set_name \(const string name\)](#)

- void `set_sort_type` (const `Sort` *`sort_type`)
- void `set_mod_type` (const `type_attr` `mod_type`)
- void `set_member_symbol` (const string `member_symbol`)
- const bool `is_synthesize` () const
- const bool `is_inherit` () const
- const string `to_string` () const
- const bool `equals` (const `Attribute` &`other`) const
- const string `key` () const

Private Member Functions

- void `copy` (const `Attribute` &`other`)
- void `destroy` ()

Private Attributes

- string `a_name`
Attribute's name.
- const `Sort` * `a_sort_type`
Attribute's type sort.
- `type_attr` `a_mod_type`
Attribute's type.
- string `a_member_symbol`
Attribute's member list.

7.2.1 Detailed Description

Definition at line 26 of file Attribute.h.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 genevalmag::Attribute::Attribute ()

Constructor empty of attribute.

Definition at line 20 of file Attribute.cpp.

7.2.2.2 genevalmag::Attribute::Attribute (const Attribute & other)

Constructor copy of attribute.

Parameters

other

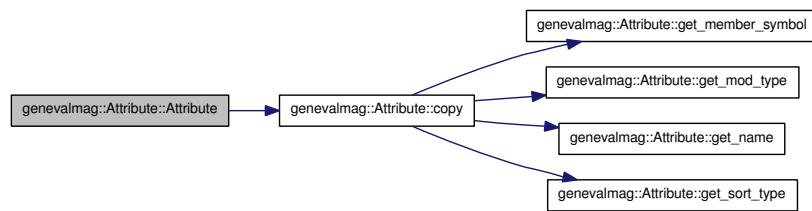
Returns

Constructor copy of attribute.

Definition at line 27 of file Attribute.cpp.

References copy().

Here is the call graph for this function:

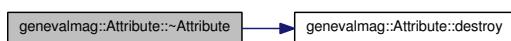
**7.2.2.3 genevalmag::Attribute::~Attribute () [virtual]**

Destructor of attribute.

Definition at line 35 of file Attribute.cpp.

References destroy().

Here is the call graph for this function:

**7.2.3 Member Function Documentation****7.2.3.1 void genevalmag::Attribute::copy (const Attribute & other) [private]**

Method of copy the attribute, STL-like C++.

Parameters

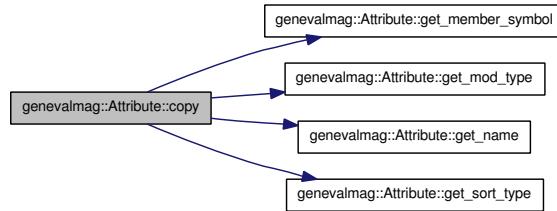
other Method of copy the attribute, STL-like.

Definition at line 56 of file Attribute.cpp.

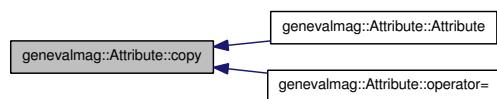
References `a_member_symbol`, `a_mod_type`, `a_name`, `a_sort_type`, `get_member_symbol()`, `get_mod_type()`, `get_name()`, and `get_sort_type()`.

Referenced by `Attribute()`, and `operator=()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.2.3.2 void genevalmag::Attribute::destroy () [private]

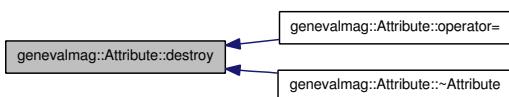
Method destroy attribute, STL-like C++.

Method destroy the attribute, STL-like.

Definition at line 67 of file Attribute.cpp.

Referenced by operator=(), and ~Attribute().

Here is the caller graph for this function:



7.2.3.3 const bool genevalmag::Attribute::equals (const Attribute & other) const

Compares the attribute with other.

Parameters

other

Returns

Compares the attribute with other.

Definition at line 181 of file Attribute.cpp.

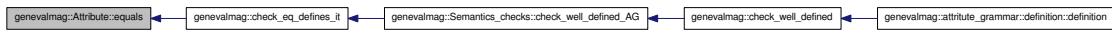
References key().

Referenced by genevalmag::check_eqDefines_it().

Here is the call graph for this function:



Here is the caller graph for this function:



7.2.3.4 const string genevalmag::Attribute::get_member_symbol () const

Returns the membership list of the attribute.

Returns

Return the membership list of the attribute.

Definition at line 98 of file Attribute.cpp.

References a_member_symbol.

Referenced by copy().

Here is the caller graph for this function:



7.2.3.5 const type_attr genevalmag::Attribute::get_mod_type () const

Returns the modifiers of the attribute.

Returns

Return the modifiers of the attribute.

Definition at line 90 of file Attribute.cpp.

References a_mod_type.

Referenced by copy().

Here is the caller graph for this function:



7.2.3.6 const string genevalmag::Attribute::get_name () const

Returns the name of the attribute.

Returns

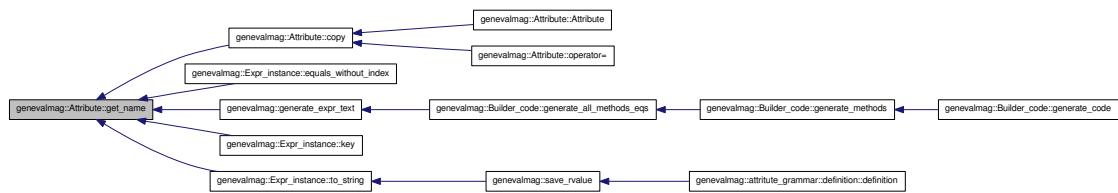
Return the name of the attribute.

Definition at line 74 of file Attribute.cpp.

References a_name.

Referenced by copy0, genevalmag::Expr_instance::equals_without_index(), genevalmag::generate_expr_text(), genevalmag::Expr_instance::key(), and genevalmag::Expr_instance::to_string().

Here is the caller graph for this function:



7.2.3.7 const Sort * genevalmag::Attribute::get_sort_type () const

Returns the sort type of the attribute.

Returns

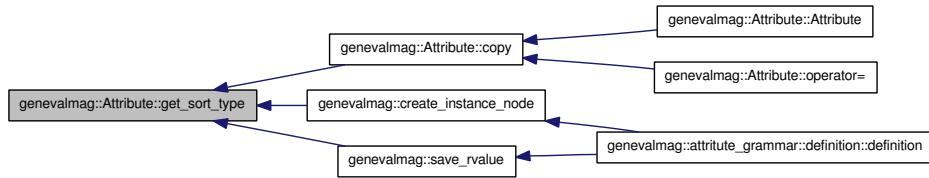
Return the sort type of the attribute.

Definition at line 82 of file Attribute.cpp.

References a_sort_type.

Referenced by copy0, genevalmag::create_instance_node(), and genevalmag::save_rvalue().

Here is the caller graph for this function:



7.2.3.8 const bool genevalmag::Attribute::is_inherit () const

Returns true if the modifiers of the attribute is inherit.

Returns

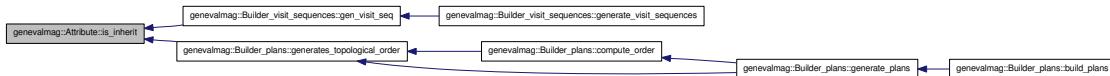
Return true if the modifiers of the attribute is inherit.

Definition at line 146 of file Attribute.cpp.

References a_mod_type, and genevalmag::k_inherit.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), genevalmag::Builder_plans::generates_topological_order(), and genevalmag::Builder_plans::generates_sequences().

Here is the caller graph for this function:



7.2.3.9 const bool genevalmag::Attribute::is_synthetize () const

Returns true if the modifiers of the attribute is synthetized.

Returns

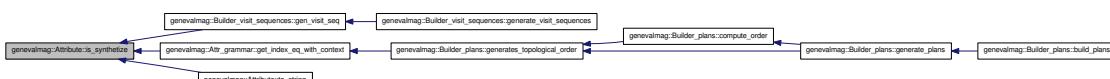
Return true if the modifiers of the attribute is synthetized.

Definition at line 138 of file Attribute.cpp.

References a_mod_type, and genevalmag::k_synthetize.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), genevalmag::Attr_grammar::get_index_eq_with_context(), and to_string().

Here is the caller graph for this function:



7.2.3.10 const string genevalmag::Attribute::key () const

Generates and returns the string key that identifies a attribute definitely.

Result = name modifiers sort_type membership_list

Returns

Generate and return the string key that identifies a attribute definitely.

Result= name modifiers sort_type membership_list

Definition at line 191 of file Attribute.cpp.

References a_member_symbol, a_name, a_sort_type, and genevalmag::Sort::get_name().

Referenced by equals().

Here is the call graph for this function:



Here is the caller graph for this function:



7.2.3.11 Attribute & genevalmag::Attribute::operator= (const Attribute & other)

Operator assign(=) of attribute.

Parameters

other

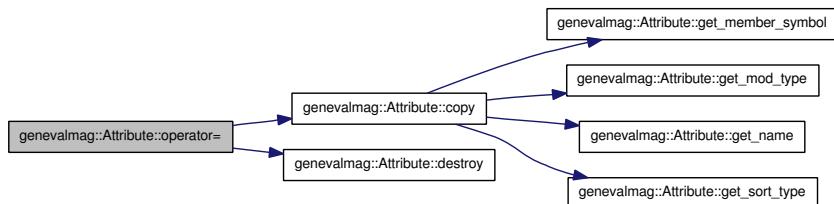
Returns

Operator assign(=) of attribute.

Definition at line 43 of file Attribute.cpp.

References copy(), and destroy().

Here is the call graph for this function:



7.2.3.12 void genevalmag::Attribute::set_member_symbol (const string *member_symbol*)

Sets the membership list of the attribute.

Parameters

member_symbol Set the membership list of the attribute.

Definition at line 130 of file Attribute.cpp.

References a_member_symbol.

Referenced by genevalmag::create_attributes().

Here is the caller graph for this function:



7.2.3.13 void genevalmag::Attribute::set_mod_type (const type_attr *mod_type*)

Sets the modifiers of the attribute.

Parameters

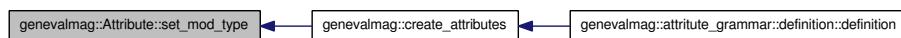
mod_type Set the modifiers of the attribute.

Definition at line 122 of file Attribute.cpp.

References a_mod_type.

Referenced by genevalmag::create_attributes().

Here is the caller graph for this function:



7.2.3.14 void genevalmag::Attribute::set_name (const string *name*)

Sets the name of the attribute.

Parameters

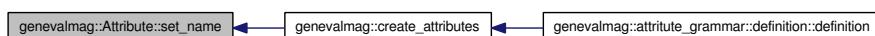
name Set the name of the attribute.

Definition at line 106 of file Attribute.cpp.

References a_name.

Referenced by genevalmag::create_attributes().

Here is the caller graph for this function:



7.2.3.15 void genevalmag::Attribute::set_sort_type (const Sort * sort_type)

Sets the sort type of the attribute.

Parameters

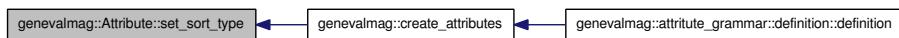
sort_type Set the sort type of the attribute.

Definition at line 114 of file Attribute.cpp.

References a_sort_type.

Referenced by genevalmag::create_attributes().

Here is the caller graph for this function:



7.2.3.16 const string genevalmag::Attribute::to_string () const

Generates and returns a string representation of a attribute.

Result = name ":" modifiers "<" sort_type ["(" instance ")" IF DEBUG IS ON] ">" of " membership_list

Returns

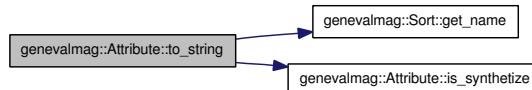
Generate and return a string representation of a attribute.

Result= name ":" modifiers "<" sort_type ["(" instance ")" IF DEBUG IS ON] ">" of " membership_list

Definition at line 156 of file Attribute.cpp.

References a_member_symbol, a_name, a_sort_type, genevalmag::Sort::get_name(), and is_synthetize().

Here is the call graph for this function:



7.2.4 Member Data Documentation

7.2.4.1 genevalmag::Attribute::a_member_symbol [private]

Attribute's member list.

Definition at line 48 of file Attribute.h.

Referenced by copy(), get_member_symbol(), key(), set_member_symbol(), and to_string().

7.2.4.2 genevalmag::Attribute::a_mod_type [private]

Attribute's type.

Definition at line 43 of file Attribute.h.

Referenced by copy(), get_mod_type(), is_inherit(), is_synthetize(), and set_mod_type().

7.2.4.3 genevalmag::Attribute::a_name [private]

Attribute's name.

Definition at line 33 of file Attribute.h.

Referenced by copy(), get_name(), key(), set_name(), and to_string().

7.2.4.4 genevalmag::Attribute::a_sort_type [private]

Attribute's type sort.

Definition at line 38 of file Attribute.h.

Referenced by copy(), get_sort_type(), key(), set_sort_type(), and to_string().

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

- include/Attr_grammar/Attribute.h
- src/Attr_grammar/Attribute.cpp

7.3 genevalmag::attritute_grammar Struct Reference

Classes

- struct definition

7.3.1 Detailed Description

Declaration of the [Attribute Grammar](#) structure with the Spirit library of Boost.

Definition at line 92 of file [Parser_AG.cpp](#).

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

- [src/Parser/Parser_AG.cpp](#)

7.4 genevalmag::attritute_grammar::definition< ScannerT > Struct Template Reference

Public Types

- `typedef rule< ScannerT > rule_exp`
- `typedef rule< typename lexeme_scanner< ScannerT >::type > rule_lexeme`

Public Member Functions

- `definition (attritute_grammar const &self)`
- `rule_exp const & start () const`

Public Attributes

- `symbols st_sorts`
- `symbols st_op_prefix`
- `symbols st_op_infix`
- `symbols st_op_postfix`
- `symbols st_functions`
- `symbols st_attributes`
- `symbols st_non_terminal`
- `rule_exp r_reserved_word`
- `rule_exp r_cpp_reserved_words`
- `rule_exp r_cpp_basic_types`
- `rule_exp r_ident`
- `rule_exp r_oper`
- `rule_exp r_char`
- `rule_exp r_string`
- `rule_exp r_boolean`
- `rule_lexeme r_id_op`
- `rule_lexeme r_string_lit`
- `rule_lexeme r_esc_seq`
- `rule_exp r_semantic_domain`
- `rule_exp r_bloq_sem`
- `rule_exp r_decl_sort`
- `rule_exp r_decl_oper`
- `rule_exp r_decl_func`
- `rule_exp r_oper_assoc`
- `rule_exp r_oper_mode`
- `rule_exp r_oper_prefix`
- `rule_exp r_oper_infix`
- `rule_exp r_oper_postfix`
- `rule_exp r_dom_func`
- `rule_exp r_attributes`
- `rule_exp r_decl_attr`
- `rule_exp r_type_attr`
- `rule_exp r_conj_symb`

- rule_exp r_rules
- rule_exp r_decl_rule
- rule_exp r_equation
- rule_exp r_right_rule
- rule_exp r_terminal
- rule_exp r_compute_eq
- rule_exp r_expression
- rule_exp r_expr_prime
- rule_exp r_expr_prime_prime
- rule_exp r_function
- rule_exp r_literal
- rule_exp r_instance
- rule_exp r_sort_st
- rule_exp r_op_prefix_st
- rule_exp r_op_infix_st
- rule_exp r_op_postfix_st
- rule_exp r_function_st
- rule_exp r_attribute_st
- rule_exp r_non_term_st
- rule_exp r_sort_stable
- rule_exp r_att_grammar

7.4.1 Detailed Description

```
template<typename ScannerT> struct genevalmag::attribute_grammar::definition<ScannerT>
```

Definition at line 95 of file Parser_AG.cpp.

7.4.2 Member Typedef Documentation

```
7.4.2.1 template<typename ScannerT> typedef rule<ScannerT>
genevalmag::attribute_grammar::definition<ScannerT>::rule_exp
```

Definition at line 354 of file Parser_AG.cpp.

```
7.4.2.2 template<typename ScannerT> typedef rule<typename lexeme_-
scanner<ScannerT>::type> genevalmag::attribute_grammar::definition<ScannerT>::rule_lexeme
```

Definition at line 357 of file Parser_AG.cpp.

7.4.3 Constructor & Destructor Documentation

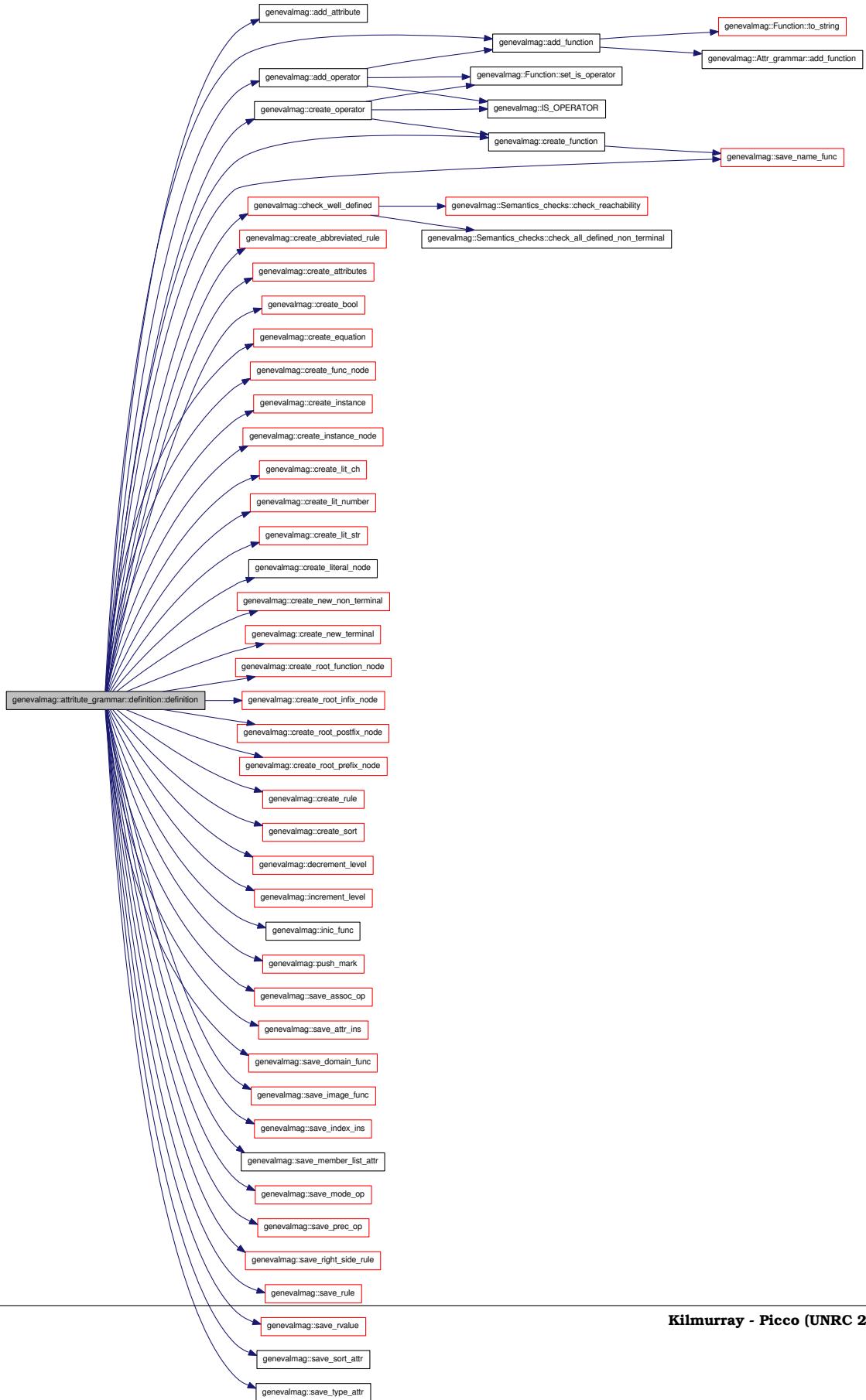
```
7.4.3.1 template<typename ScannerT> genevalmag::attribute_grammar::definition<ScannerT>::definition(attribute_grammar const & self) [inline]
```

Definition at line 97 of file Parser_AG.cpp.

References genevalmag::add_attribute(), genevalmag::add_function(), genevalmag::add_operator(), genevalmag::check_well_defined(), genevalmag::create_abbreviated_rule(), genevalmag::create_attributes(), genevalmag::create_bool(), genevalmag::create_equation(), genevalmag::create_func_node(), genevalmag::create_function(), genevalmag::create_instance(), genevalmag::create_instance_node(), genevalmag::create_lit_ch(), genevalmag::create_lit_number(), genevalmag::create_lit_str(), genevalmag::create_literal_node(), genevalmag::create_new_non_terminal(), genevalmag::create_new_terminal(), genevalmag::create_operator(), genevalmag::create_root_function_node(), genevalmag::create_root_infix_node(), genevalmag::create_root_postfix_node(), genevalmag::create_root_prefix_node(), genevalmag::create_rule(), genevalmag::create_sort(), genevalmag::decrement_level(), genevalmag::increment_level(), genevalmag::inic_func(), genevalmag::push_mark(), genevalmag::attribute_grammar::definition< ScannerT >::r_att_grammar, genevalmag::attribute_grammar::definition< ScannerT >::r_attribute_st, genevalmag::attribute_grammar::definition< ScannerT >::r_attributes, genevalmag::attribute_grammar::definition< ScannerT >::r_bloc_sem, genevalmag::attribute_grammar::definition< ScannerT >::r_boolean, genevalmag::attribute_grammar::definition< ScannerT >::r_char, genevalmag::attribute_grammar::definition< ScannerT >::r_compute_eq, genevalmag::attribute_grammar::definition< ScannerT >::r_conj_symb, genevalmag::attribute_grammar::definition< ScannerT >::r_cpp_basic_types, genevalmag::attribute_grammar::definition< ScannerT >::r_cpp_reserved_words, genevalmag::attribute_grammar::definition< ScannerT >::r_decl_attr, genevalmag::attribute_grammar::definition< ScannerT >::r_decl_func, genevalmag::attribute_grammar::definition< ScannerT >::r_decl_oper, genevalmag::attribute_grammar::definition< ScannerT >::r_decl_rule, genevalmag::attribute_grammar::definition< ScannerT >::r_decl_sort, genevalmag::attribute_grammar::definition< ScannerT >::r_dom_func, genevalmag::attribute_grammar::definition< ScannerT >::r_equation, genevalmag::attribute_grammar::definition< ScannerT >::r_esc_seq, genevalmag::attribute_grammar::definition< ScannerT >::r_expr_prime, genevalmag::attribute_grammar::definition< ScannerT >::r_expr_prime_prime, genevalmag::attribute_grammar::definition< ScannerT >::r_expression, genevalmag::attribute_grammar::definition< ScannerT >::r_function, genevalmag::attribute_grammar::definition< ScannerT >::r_function_st, genevalmag::attribute_grammar::definition< ScannerT >::r_id_op, genevalmag::attribute_grammar::definition< ScannerT >::r_ident, genevalmag::attribute_grammar::definition< ScannerT >::r_instance, genevalmag::attribute_grammar::definition< ScannerT >::r_literal, genevalmag::attribute_grammar::definition< ScannerT >::r_non_term_st, genevalmag::attribute_grammar::definition< ScannerT >::r_op_infix_st, genevalmag::attribute_grammar::definition< ScannerT >::r_op_postfix_st, genevalmag::attribute_grammar::definition< ScannerT >::r_op_prefix_st, genevalmag::attribute_grammar::definition< ScannerT >::r_oper, genevalmag::attribute_grammar::definition< ScannerT >::r_oper_assoc, genevalmag::attribute_grammar::definition< ScannerT >::r_oper_infix, genevalmag::attribute_grammar::definition< ScannerT >::r_oper_mode, genevalmag::attribute_grammar::definition< ScannerT >::r_oper_postfix, genevalmag::attribute_grammar::definition< ScannerT >::r_oper_prefix, genevalmag::attribute_grammar::definition< ScannerT >::r_reserved_word, genevalmag::attribute_grammar::definition< ScannerT >::r_right_rule, genevalmag::attribute_grammar::definition< ScannerT >::r_rules, genevalmag::attribute_grammar::definition< ScannerT >::r_semantic_domain, genevalmag::attribute_grammar::definition< ScannerT >::r_sort_st, genevalmag::attribute_grammar::definition< ScannerT >::r_string, genevalmag::attribute_grammar::definition< ScannerT >::r_string_lit, genevalmag::attribute_grammar::definition< ScannerT >::r_terminal, genevalmag::attribute_grammar::definition< ScannerT >::r_type_attr, genevalmag::save_assoc_op0, genevalmag::save_attr_ins(), genevalmag::save_domain_func(), genevalmag::save_image_func(), genevalmag::save_index_ins(), genevalmag::save_member_list_attr0,

genevalmag::save_mode_op(), genevalmag::save_name_func(), genevalmag::save_prec_op(), genevalmag::save_right_side_rule(), genevalmag::save_rule(), genevalmag::save_rvalue(), genevalmag::save_sort_attr(), genevalmag::save_type_attr(), genevalmag::attritute_grammar::definition< ScannerT >::st_attributes, genevalmag::attritute_grammar::definition< ScannerT >::st_functions, genevalmag::attritute_grammar::definition< ScannerT >::st_non_terminal, genevalmag::attritute_grammar::definition< ScannerT >::st_op_infix, genevalmag::attritute_grammar::definition< ScannerT >::st_op_postfix, genevalmag::attritute_grammar::definition< ScannerT >::st_op_prefix, and genevalmag::attritute_grammar::definition< ScannerT >::st_sorts.

Here is the call graph for this function:



7.4.4 Member Function Documentation

7.4.4.1 template<typename ScannerT> rule_exp const& genevalmag::attribute_grammar::definition<ScannerT>::start() const [inline]

Definition at line 388 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition<ScannerT>::r_att_grammar.

7.4.5 Member Data Documentation

7.4.5.1 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition<ScannerT>::r_att_grammar

Definition at line 386 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition<ScannerT>::definition(), and genevalmag::attribute_grammar::definition<ScannerT>::start().

7.4.5.2 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition<ScannerT>::r_attribute_st

Definition at line 382 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition<ScannerT>::definition().

7.4.5.3 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition<ScannerT>::r_attributes

Definition at line 371 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition<ScannerT>::definition().

7.4.5.4 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition<ScannerT>::r_bloc_sem

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition<ScannerT>::definition().

7.4.5.5 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition<ScannerT>::r_boolean

Definition at line 360 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition<ScannerT>::definition().

7.4.5.6 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition<ScannerT>::r_char

Definition at line 360 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

7.4.5.7 template<typename ScannerT > rule_exp genevalmag::attritute_grammar::definition< ScannerT >::r_compute_eq

Definition at line 374 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

7.4.5.8 template<typename ScannerT > rule_exp genevalmag::attritute_grammar::definition< ScannerT >::r_conj_symb

Definition at line 371 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

7.4.5.9 template<typename ScannerT > rule_exp genevalmag::attritute_grammar::definition< ScannerT >::r_cpp_basic_types

Definition at line 360 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

7.4.5.10 template<typename ScannerT > rule_exp genevalmag::attritute_grammar::definition< ScannerT >::r_cpp_reserved_words

Definition at line 360 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

7.4.5.11 template<typename ScannerT > rule_exp genevalmag::attritute_grammar::definition< ScannerT >::r_decl_attr

Definition at line 371 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

7.4.5.12 template<typename ScannerT > rule_exp genevalmag::attritute_grammar::definition< ScannerT >::r_decl_func

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

7.4.5.13 template<typename ScannerT > rule_exp genevalmag::attritute_grammar::definition< ScannerT >::r_decl_oper

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

7.4.5.14 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_decl_rule

Definition at line 374 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.15 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_decl_sort

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.16 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_dom_func

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.17 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_equation

Definition at line 374 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.18 template<typename ScannerT > rule_lexeme genevalmag::attribute_grammar::definition< ScannerT >::r_esc_seq

Definition at line 363 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.19 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_expr_prime

Definition at line 378 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.20 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_expr_prime_prime

Definition at line 378 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.21 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_expression

Definition at line 378 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.22 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_function

Definition at line 378 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.23 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_function_st

Definition at line 382 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.24 template<typename ScannerT > rule_lexeme genevalmag::attribute_grammar::definition< ScannerT >::r_id_op

Definition at line 363 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.25 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_ident

Definition at line 360 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.26 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_instance

Definition at line 378 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.27 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_literal

Definition at line 378 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.28 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_non_term_st

Definition at line 382 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.29 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_op_infix_st

Definition at line 382 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.30 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_op_postfix_st

Definition at line 382 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.31 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_op_prefix_st

Definition at line 382 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.32 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_oper

Definition at line 360 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.33 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_oper_assoc

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.34 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_oper_infix

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.35 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_oper_mode

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.36 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_oper_postfix

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.37 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_oper_prefix

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.38 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_reserved_word

Definition at line 360 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.39 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_right_rule

Definition at line 374 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.40 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_rules

Definition at line 374 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.41 template<typename ScannerT > rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_semantic_domain

Definition at line 366 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.42 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_sort_st

Definition at line 382 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.43 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_sort_stable

Definition at line 382 of file Parser_AG.cpp.

7.4.5.44 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_string

Definition at line 360 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.45 template<typename ScannerT> rule_lexeme genevalmag::attribute_grammar::definition< ScannerT >::r_string_lit

Definition at line 363 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.46 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_terminal

Definition at line 374 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.47 template<typename ScannerT> rule_exp genevalmag::attribute_grammar::definition< ScannerT >::r_type_attr

Definition at line 371 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.48 template<typename ScannerT> symbols genevalmag::attribute_grammar::definition< ScannerT >::st_attributes

Definition at line 348 of file Parser_AG.cpp.

Referenced by genevalmag::attribute_grammar::definition< ScannerT >::definition().

7.4.5.49 template<typename ScannerT> symbols genevalmag::attribute_grammar::definition< ScannerT >::st_functions

Definition at line 347 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

**7.4.5.50 template<typename ScannerT > symbols genevalmag::attritute_-
grammar::definition< ScannerT >::st_non_terminal**

Definition at line 349 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

**7.4.5.51 template<typename ScannerT > symbols genevalmag::attritute_-
grammar::definition< ScannerT >::st_op_infix**

Definition at line 345 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

**7.4.5.52 template<typename ScannerT > symbols genevalmag::attritute_-
grammar::definition< ScannerT >::st_op_postfix**

Definition at line 346 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

**7.4.5.53 template<typename ScannerT > symbols genevalmag::attritute_-
grammar::definition< ScannerT >::st_op_prefix**

Definition at line 344 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

**7.4.5.54 template<typename ScannerT > symbols genevalmag::attritute_-
grammar::definition< ScannerT >::st_sorts**

Definition at line 343 of file Parser_AG.cpp.

Referenced by genevalmag::attritute_grammar::definition< ScannerT >::definition().

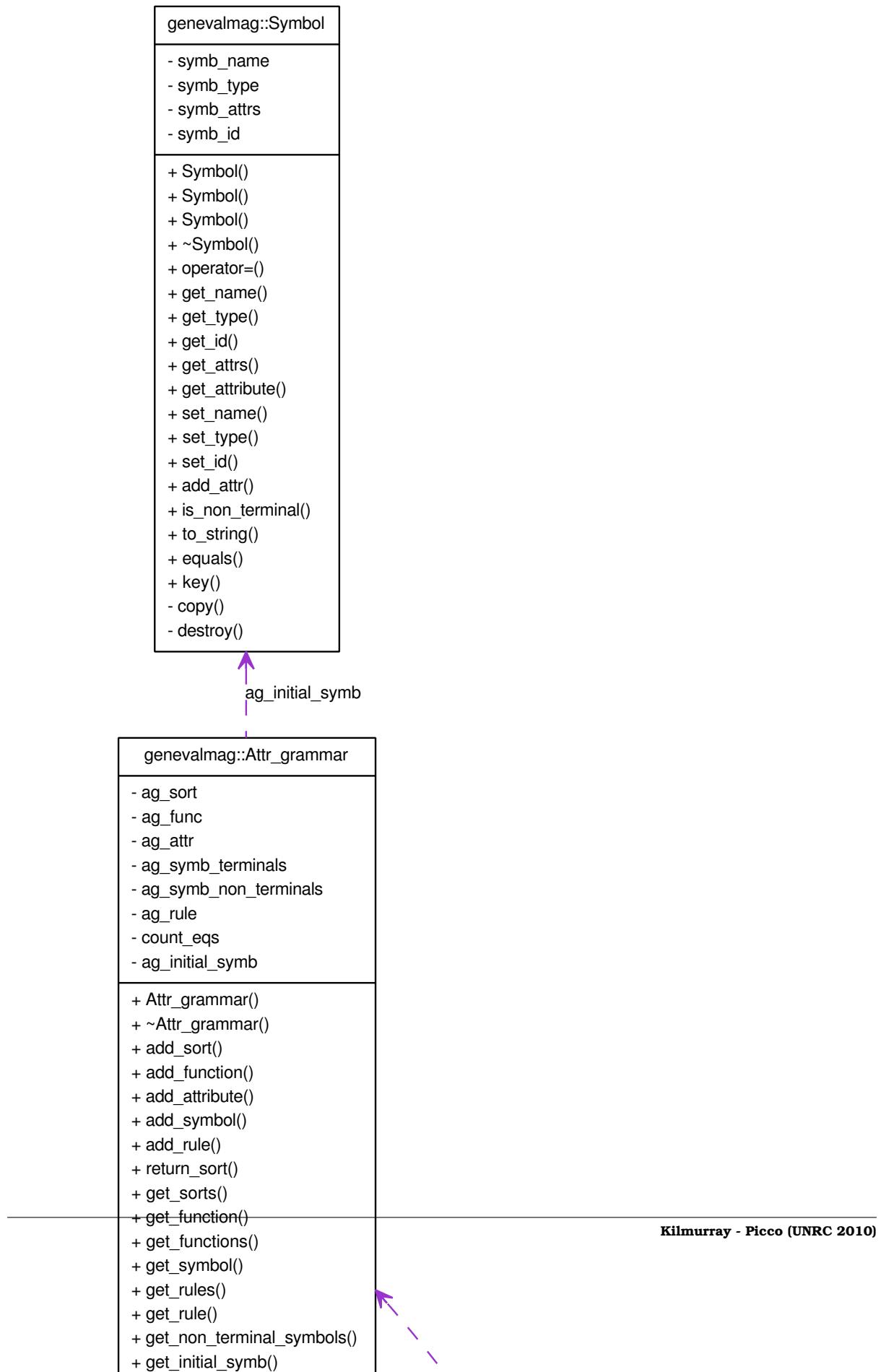
The documentation for this struct was generated from the following file:

- src/Parser/[Parser_AG.cpp](#)

7.5 genevalmag::Builder_code Class Reference

```
#include <Builder_code.h>
```

Collaboration diagram for genevalmag::Builder_code:



Public Member Functions

- `Builder_code (const string &path_folder_output, const string &name_file, const Attr_grammar &attribute_grammar, const Builder_plans &builder_plan, const Builder_visit_sequences &builder_v_seq)`
- `virtual ~Builder_code ()`
- `bool generate_code (const vector< string > &headers_file, const string path) const`

Private Member Functions

- `void generate_header_file () const`
- `void generate_footer_header () const`
- `void generate_code_file (const vector< string > &headers_file) const`
- `void generate_structs () const`
- `void generate_constructor () const`
- `void generate_methods () const`
- `void generate_footer_code () const`
- `string generate_print () const`
- `string generate_translate () const`
- `string generate_traverse () const`
- `string generate_all_compute_eq () const`
- `string generate_evaluator () const`
- `string generate_eval_visiter () const`
- `string generate_add_plan () const`
- `string generate_add_plan_project () const`
- `string generate_all_methods_eqs () const`
- `string generate_initialize_rules () const`
- `string generate_initialize_v_seq () const`
- `string generate_initialize_context () const`
- `string generate_initialize_plans () const`
- `string generate_initialize_plan_proj () const`

Private Attributes

- `const Attr_grammar & attr_grammar`
References to the attribute grammar.
- `const Builder_plans & b_plans`
References to all plans generates.
- `const Builder_visit_sequences & b_v_seq`
References to all visit sequences generates.
- `string path_output`
Defines the path where the files generated be saved.
- `string file_name`
Defines the name of the files and the evaluator's class name.

7.5.1 Detailed Description

Definition at line 17 of file Builder_code.h.

7.5.2 Constructor & Destructor Documentation

7.5.2.1 `genevalmag::Builder_code::Builder_code (const string & path_folder_output, const string & name_file, const Attr_grammar & attribute_grammar, const Builder_plans & builder_plan, const Builder_visit_sequences & builder_v_seq)`

Constructor with the path and name of output files to be generated.

Parameters

path_folder_output
name_file
attribute_grammar
builder_plan
builder_v_seq

Returns

Constructor with the path and name of output files to be generated.

Definition at line 33 of file Builder_code.cpp.

7.5.2.2 `genevalmag::Builder_code::~Builder_code () [virtual]`

Destructor of the [Builder_code](#).

Returns

Destructor of the [Builder_code](#).

Definition at line 52 of file Builder_code.cpp.

7.5.3 Member Function Documentation

7.5.3.1 `string genevalmag::Builder_code::generate_add_plan () const [private]`

Generates and inserts the class method that insert a new plan.

Returns

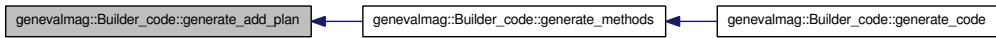
Generates and inserts the class method that insert a new plan.

Definition at line 888 of file Builder_code.cpp.

References file_name.

Referenced by generate_methods0.

Here is the caller graph for this function:



7.5.3.2 string genevalmag::Builder_code::generate_add_plan_project () const [private]

Generates and inserts the class method that insert a new projected plan.

Returns

Generates and inserts the class method that insert a new projected plan.

Definition at line 903 of file Builder_code.cpp.

References file_name.

Referenced by generate_methods0.

Here is the caller graph for this function:



7.5.3.3 string genevalmag::Builder_code::generate_all_compute_eq () const [private]

Generates a method with a large switch with all equations, invoking in each case, the method that computes.

Returns

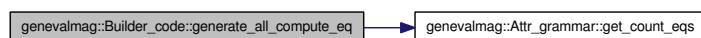
Generates a method with a large switch with all equations, invoking in each case, the method that computes.

Definition at line 806 of file Builder_code.cpp.

References attr_grammar, file_name, and genevalmag::Attr_grammar::get_count_eqs0.

Referenced by generate_methods0.

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.4 string genevalmag::Builder_code::generate_all_methods_eqs () const [private]

Generates one method for each equation in the grammar, that computes it's value.

Returns

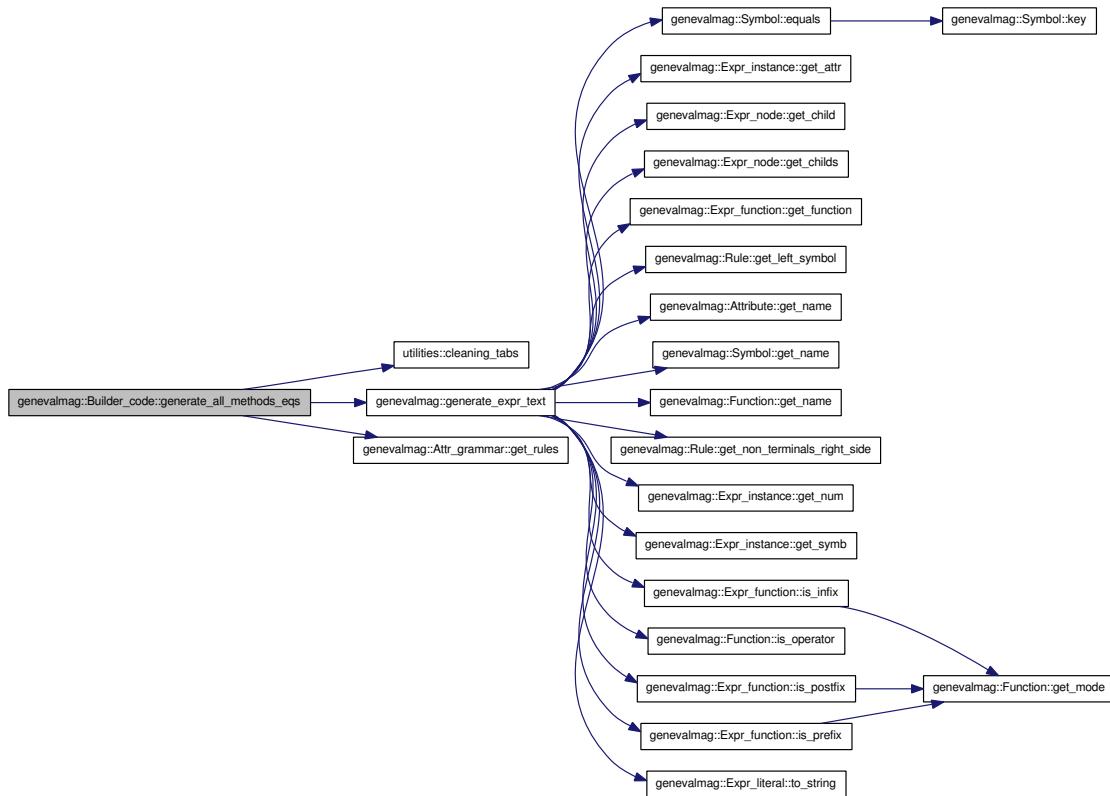
Generates one method for each equation in the grammar, that computes it's value.

Definition at line 736 of file Builder_code.cpp.

References attr_grammar, utilities::cleaning_tabs(), genevalmag::generate_expr_text(), and genevalmag::Attr_grammar::get_rules().

Referenced by generate_methods().

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.5 `bool genevalmag::Builder_code::generate_code (const vector< string > & headers_file, const string path) const`

Generates the header and source code of the static evaluator of the grammar passed as parameter, with their evaluations plans, visit sequence and headers for uses user functions defined.

Parameters

`headers_file`

`path`

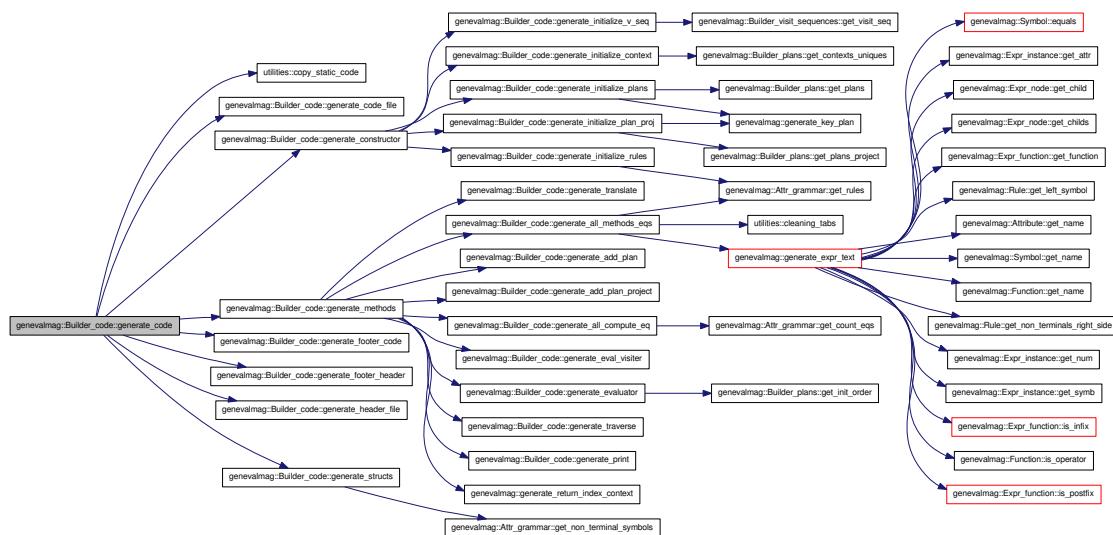
Returns

Generates the header and source code of the static evaluator of the grammar passed as parameter, alog with their evaluations plans, visit sequence and headers for uses user functions defined.

Definition at line 1096 of file Builder_code.cpp.

References `utilities::copy_static_code()`, `generate_code_file()`, `generate_constructor()`, `generate_footer_code()`, `generate_footer_header()`, `generate_header_file()`, `generate_methods()`, `generate_structs()`, and `path_output`.

Here is the call graph for this function:



7.5.3.6 void genevalmag::Builder_code::generate_code_file (const vector< string > & headers_file) const [private]

Create the source code file of the evaluator with some information.

Parameters

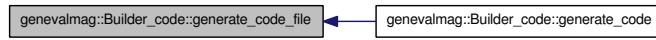
headers_file Create the source code file of the evaluator with some information.

Definition at line 111 of file Builder_code.cpp.

References file_name, and path_output.

Referenced by generate_code().

Here is the caller graph for this function:



7.5.3.7 void genevalmag::Builder_code::generate_constructor () const [private]

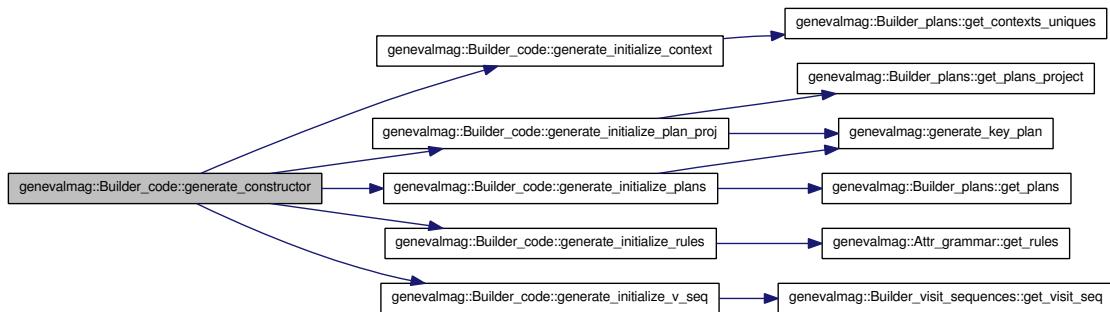
Generates and inserts the evaluator class's constructor. With all initializations of Evaluation Plans, Evaluation Plans Project, Visit Sequences and Rules.

Definition at line 479 of file Builder_code.cpp.

References file_name, generate_initialize_context(), generate_initialize_plan_proj(), generate_initialize_plans(), generate_initialize_rules(), generate_initialize_v_seq(), and path_output.

Referenced by generate_code().

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.8 string genevalmag::Builder_code::generate_eval_visiter () const [private]

Generates the evaluator method, which following the visit sequences drawn, visit the nodes of the tree until computes it completely.

Returns

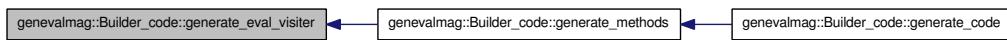
Generates the evaluator method, which following the visit sequences drawn, visit the nodes of the tree until computes it completely.

Definition at line 834 of file Builder_code.cpp.

References file_name.

Referenced by generate_methods().

Here is the caller graph for this function:

**7.5.3.9 string genevalmag::Builder_code::generate_evaluator () const [private]**

Generates the evaluating method, which performs the method invocations to be computed all the attributes of the AST.

Returns

Generates the evaluating method, which performs the method invocations to be computed all the attributes of the AST.

Definition at line 866 of file Builder_code.cpp.

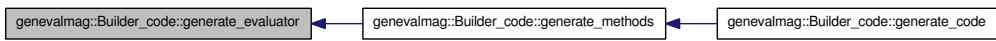
References b_plans, file_name, and genevalmag::Builder_plans::get_init_order().

Referenced by generate_methods().

Here is the call graph for this function:



Here is the caller graph for this function:

**7.5.3.10 void genevalmag::Builder_code::generate_footer_code () const [private]**

Insert the source code file's footer of the evaluator.

Definition at line 210 of file Builder_code.cpp.

References file_name, and path_output.

Referenced by generate_code().

Here is the caller graph for this function:



7.5.3.11 void genevalmag::Builder_code::generate_footer_header () const [private]

Insert the header file's footer of the evaluator.

Definition at line 162 of file Builder_code.cpp.

References file_name, and path_output.

Referenced by generate_code().

Here is the caller graph for this function:



7.5.3.12 void genevalmag::Builder_code::generate_header_file () const [private]

Create the header file of the evaluator with some information.

Definition at line 59 of file Builder_code.cpp.

References file_name, and path_output.

Referenced by generate_code().

Here is the caller graph for this function:



7.5.3.13 string genevalmag::Builder_code::generate_initialize_context () const [private]

Generates the initialization of all contexts rules uniques.

Returns

Generates the initialization of all contexts rules uniques.

Definition at line 330 of file Builder_code.cpp.

References b_plans, and genevalmag::Builder_plans::get_contexts_uniques().

Referenced by generate_constructor().

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.14 string genevalmag::Builder_code::generate_initialize_plan_proj () const [private]

Generates the initialization of all evaluation plan projects.

Returns

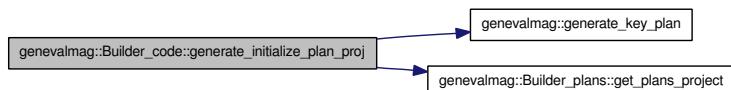
Generates the initialization of all evaluation plan projects.

Definition at line 388 of file Builder_code.cpp.

References b_plans, genevalmag::generate_key_plan(), and genevalmag::Builder_plans::get_plans_project().

Referenced by generate_constructor().

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.15 string genevalmag::Builder_code::generate_initialize_plans () const [private]

Generates the initialization of all evaluations plans.

Returns

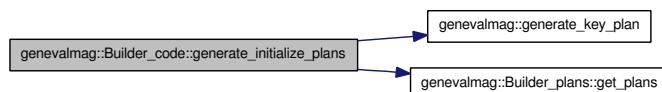
Generates the initialization of all evaluations plans.

Definition at line 356 of file Builder_code.cpp.

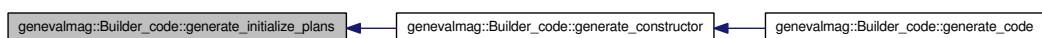
References b_plans, genevalmag::generate_key_plan(), and genevalmag::Builder_plans::get_plans().

Referenced by generate_constructor().

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.16 string genevalmag::Builder_code::generate_initialize_rules () const [private]

Generates the initialization of all rules.

Returns

Generates the initialization of all rules.

Definition at line 435 of file Builder_code.cpp.

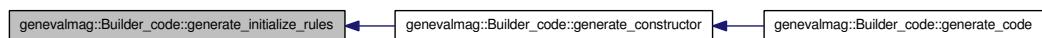
References attr_grammar, and genevalmag::Attr_grammar::get_rules().

Referenced by generate_constructor().

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.17 string genevalmag::Builder_code::generate_initialize_v_seq () const [private]

Generates the initialization of all visit sequences.

Returns

Generates the initialization of all visit sequences.

Definition at line 280 of file Builder_code.cpp.

References b_v_seq, and genevalmag::Builder_visit_sequences::get_visit_seq().

Referenced by generate_constructor().

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.18 void genevalmag::Builder_code::generate_methods () const [private]

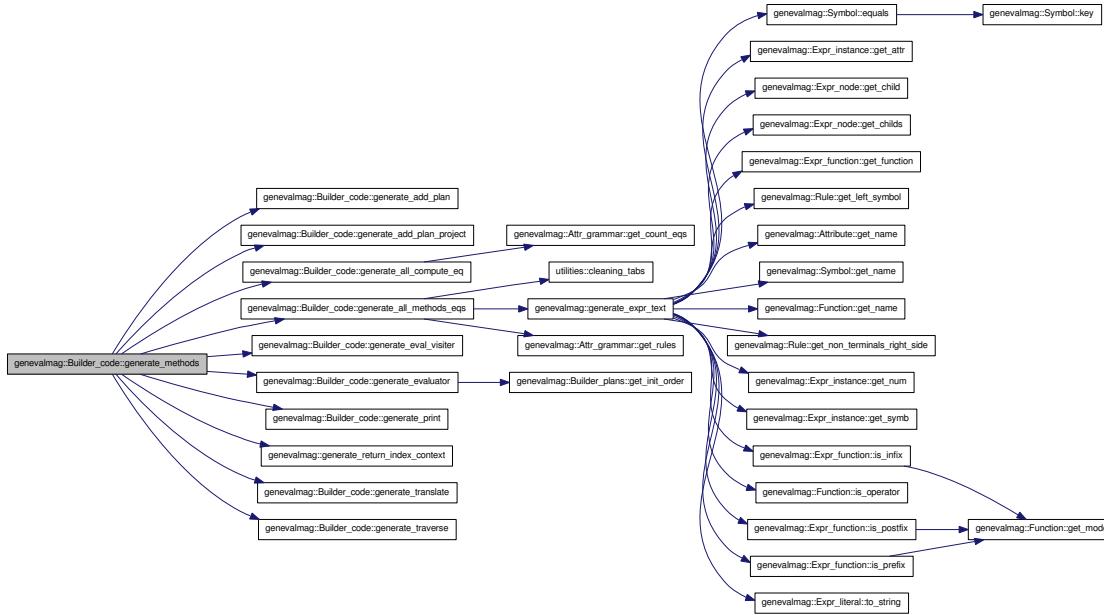
Generates and inserts all class methods, includind traverse, visit evaluator and the main evaluator. This methods are based on the article by Wuu Yang.

Definition at line 919 of file Builder_code.cpp.

References file_name, generate_add_plan(), generate_add_plan_project(), generate_all_compute_eq(), generate_all_methods_eqs(), generate_eval_visiter(), generate_evaluator(), generate_print(), genevalmag::generate_return_index_context(), generate_translate(), generate_traverse(), and path_output.

Referenced by generate_code().

Here is the call graph for this function:



Here is the caller graph for this function:



7.5.3.19 string genevalmag::Builder_code::generate_print () const [private]

Generates the print method, for show the all visit sequences.

Returns

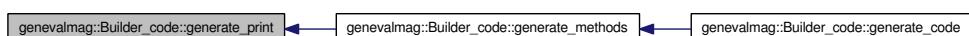
Generates the print method, for show the all visit sequences.

Definition at line 509 of file Builder_code.cpp.

References file_name.

Referenced by generate_methods().

Here is the caller graph for this function:



7.5.3.20 void genevalmag::Builder_code::generate_structs () const [private]

Generates and inserts all structs for represent each symbol of the grammar. With constructor and to_string methods.

For symbol S, with attrs s1 <syn> int, inserts:

```
typedef struct Symbol_S: Node
{
    int s0;
    Symbol_S(unsigned short r_id);
    string to_string() const;
} S;
```

Generates and inserts all structs for represent each symbol of the grammar. With constructor and to_string methods.

For symbol S, with attrs s1 <syn> int, inserts:

```
typedef struct Symbol_S: Node { int s0;
    Symbol_S(unsigned short r_id);
    string to_string() const; } S;
```

Definition at line 966 of file Builder_code.cpp.

References attr_grammar, file_name, genevalmag::Attr_grammar::get_non_terminal_symbols(), and path_output.

Referenced by generate_code().

Here is the call graph for this function:



Here is the caller graph for this function:

**7.5.3.21 string genevalmag::Builder_code::generate_translate () const [private]**

Generates the print method, for show the all visit sequences in a format more descriptive.

Returns

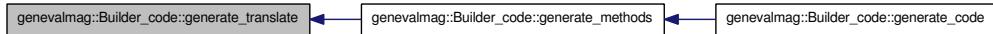
Generates the print method, for show the all visit sequences in a format more descriptive.

Definition at line 534 of file Builder_code.cpp.

References file_name.

Referenced by generate_methods0.

Here is the caller graph for this function:



7.5.3.22 string genevalmag::Builder_code::generate_traverse () const [private]

Generates the method that crosses the AST and sets the evaluation plan that corresponds to each node.

Returns

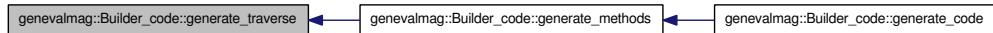
Generates the method that crosses the AST and sets the evaluation plan that corresponds to each node.

Definition at line 588 of file Builder_code.cpp.

References file_name.

Referenced by generate_methods0.

Here is the caller graph for this function:



7.5.4 Member Data Documentation

7.5.4.1 genevalmag::Builder_code::attr_grammar [private]

References to the attribute grammar.

Definition at line 21 of file Builder_code.h.

Referenced by generate_all_compute_eq0(), generate_all_methods_eqs0(), generate_initialize_rules0(), and generate_structs0().

7.5.4.2 genevalmag::Builder_code::b_plans [private]

References to all plans generates.

Definition at line 27 of file Builder_code.h.

Referenced by generate_evaluator0(), generate_initialize_context0(), generate_initialize_plan_proj0(), and generate_initialize_plans0().

7.5.4.3 genevalmag::Builder_code::b_v_seq [private]

References to all visit sequences generates.

Definition at line 33 of file Builder_code.h.

Referenced by generate_initialize_v_seq0.

7.5.4.4 **genevalmag::Builder_code::file_name** [private]

Defines the name of the files and the evaluator's class name.

Definition at line 45 of file Builder_code.h.

Referenced by generate_add_plan(), generate_add_plan_project(), generate_all_compute_eq(), generate_code_file(), generate_constructor(), generate_eval_visiter(), generate_evaluator(), generate_footer_code(), generate_footer_header(), generate_header_file(), generate_methods(), generate_print(), generate_structs(), generate_translate(), and generate_traverse().

7.5.4.5 **genevalmag::Builder_code::path_output** [private]

Defines the path where the files generated be saved.

Definition at line 39 of file Builder_code.h.

Referenced by generate_code(), generate_code_file(), generate_constructor(), generate_footer_code(), generate_footer_header(), generate_header_file(), generate_methods(), and generate_structs().

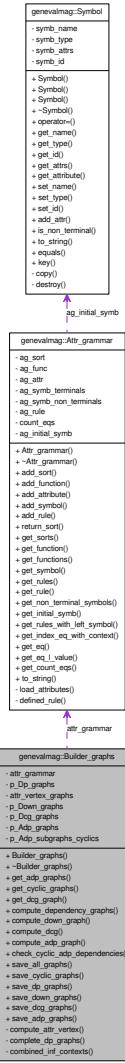
The documentation for this class was generated from the following files:

- include/Builders/[Builder_code.h](#)
- src/Builders/[Builder_code.cpp](#)

7.6 genevalmag::Builder_graphs Class Reference

```
#include <Builder_graphs.h>
```

Collaboration diagram for genevalmag::Builder_graphs:



Public Member Functions

- `Builder_graphs (const Attr_grammar &attribute_grammar)`
- `~Builder_graphs ()`
- `const map<vector<unsigned short>, Graph> &get_adp_graphs () const`
- `const map<vector<unsigned short>, Graph> &get_cyclic_graphs () const`
- `const Graph &get_dcg_graph (unsigned short index_rule) const`
- `bool compute_dependency_graphs ()`
- `bool compute_down_graph ()`
- `bool compute_dcg ()`
- `bool compute_adp_graph ()`

- bool `check_cyclic_adp_dependencies ()`
- bool `save_all_graphs (const string path_output) const`
- bool `save_cyclic_graphs (const string path_output) const`
- bool `save_dp_graphs (const string path_output) const`
- bool `save_down_graphs (const string path_output) const`
- bool `save_dcg_graphs (const string path_output) const`
- bool `save_adp_graphs (const string path_output) const`

Private Member Functions

- void `compute_attr_vertex ()`
- void `complete_dp_graphs ()`
- bool `combined_inf_contexts (const Rule *rule, Graph &graph, vector< vector< unsigned short > > &inf_context, size_t index_to_combine)`

Private Attributes

- const Attr_grammar & attr_grammar
References to the attribute grammar.
- map< unsigned short, Graph > p_Dp_graphs
Store the DP graphs. The key corresponds to the key Rule..
- map< string, Graph > attr_vertex_graphs
Store the vertex-attr graphs. The key corresponds to the key Symbol.
- map< string, Graph > p_Down_graphs
Store the down graphs. The key corresponds to the key Symbol.
- map< unsigned short, Graph > p_Dcg_graphs
Store the dcg graphs. The key corresponds to the key Rule.
- map< vector< unsigned short >, Graph > p_Adp_graphs
Store the adp graphs. The key corresponds to the key Rule with the inferior context.
- map< vector< unsigned short >, Graph > p_Adp_subgraphs_cyclics
Store the adp graphs. The key corresponds to the key Rule with the inferior context.

7.6.1 Detailed Description

Definition at line 40 of file Builder_graphs.h.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 `genevalmag::Builder_graphs::Builder_graphs (const Attr_grammar & attribute_grammar)`

Constructor empty of Builder graphs.

Parameters

attribute_grammar

Returns

Constructor empty of Builder graphs.

Definition at line 56 of file Builder_graphs.cpp.

7.6.2.2 `genevalmag::Builder_graphs::~Builder_graphs ()`

Destructor empty of Builder graphs.

Returns

Destructor empty of Builder graphs.

Definition at line 63 of file Builder_graphs.cpp.

7.6.3 Member Function Documentation

7.6.3.1 `bool genevalmag::Builder_graphs::check_cyclic_adp_dependencies ()`

Checks if the graph contains cycle. Utilities a depth-first-search for traverse the graphs.

Returns

Checks if the graph contains cycle. Utilities a depth-first-search for traverse the graphs.

Definition at line 444 of file Builder_graphs.cpp.

References p_Adp_graphs, and p_Adp_subgraphs_cyclics.

Referenced by genevalmag::Builder_plans::build_plans().

Here is the caller graph for this function:



7.6.3.2 bool genevalmag::Builder_graphs::combined_inf_contexts (const Rule * *rule*, Graph & *graph*, vector<vector<unsigned short>> & *inf_context*, size_t *index_to_combine*) [private]

Generate all combinations of the rules and saves a graph ADP for each of them.

Parameters

- rule*
- graph*
- inf_context*
- index_to_combine*

Returns

Generate all combinations of the rules and saves a graph ADP for each of them.

Definition at line 301 of file Builder_graphs.cpp.

References genevalmag::Rule::key(), utilities::merge_graph(), p_Adp_graphs, and p_Dcg_graphs.

Referenced by compute_adp_graph().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.3 void genevalmag::Builder_graphs::complete_dp_graphs () [private]

Completes dp-graph with the vertex of low on instances.

Definition at line 469 of file Builder_graphs.cpp.

References attr_grammar, attr_vertex_graphs, genevalmag::Attr_grammar::get_rules(), utilities::merge_graph(), and p_Dp_graphs.

Referenced by compute_down_graph().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.4 bool genevalmag::Builder_graphs::compute_adp_graph ()

Algorithm ADP Builds the graphs ADP for each rule of the grammar. Ex: ADP(1): (1) E:= E + T. graph G: DP(1) U Dcg E (J1..JN) U Dcg T (K1..KM) Where Ji y ki are rule with left-symbol E and T respectively.

Returns

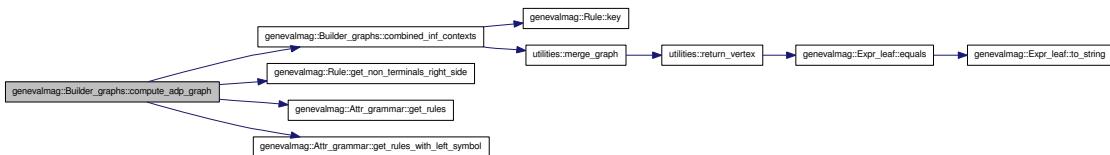
Algorithm ADP Builds the graphs ADP for each rule of the grammar. Ex: ADP(1): (1) E:= E + T. graph G: DP(1) U Dcg E (J1..JN) U Dcg T (K1..KM) Where Ji y ki are rule with left-symbol E and T respectively.

Definition at line 360 of file Builder_graphs.cpp.

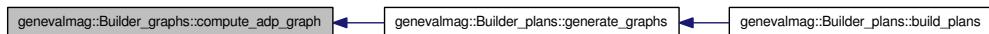
References attr_grammar, combined_inf_contexts(), genevalmag::current_rule, genevalmag::Rule::get_non_terminals_right_side(), genevalmag::Attr_grammar::get_rules(), genevalmag::Attr_grammar::get_rules_with_left_symbol(), and p_Dp_graphs.

Referenced by genevalmag::Builder_plans::generate_graphs().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.5 void genevalmag::Builder_graphs::compute_attr_vertex () [private]

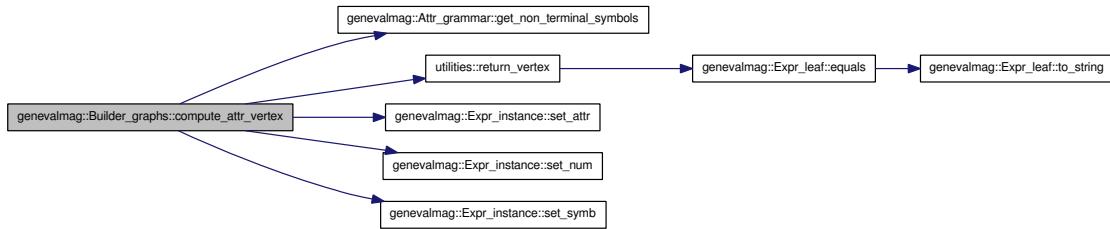
Builds a graph for each symbol of the grammar with all attributes. The graph only has vertexes. It hasn't edges. Ex: Symbol E ; attributes: s,i graph: vertex: E.s, E.i

Definition at line 155 of file Builder_graphs.cpp.

References attr_grammar, attr_vertex_graphs, genevalmag::Attr_grammar::get_non_terminal_symbols(), p_Dp_graphs, utilities::return_vertex(), genevalmag::Expr_instance::set_attr(), genevalmag::Expr_instance::set_num(), and genevalmag::Expr_instance::set_symb().

Referenced by compute_down_graph().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.6 bool genevalmag::Builder_graphs::compute_dcg ()

Algorithm DCG Builds the graph Dcg for each rule of the grammar. Ex: Dcg E(1) : (1) E:= E + T. graph G: DP(1) U Down(E) U Down(T) Project(G,{attributes of E})

Returns

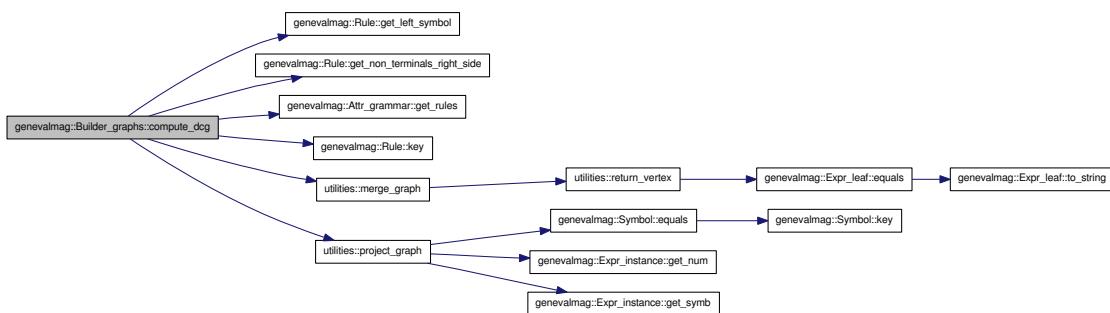
Algorithm DCG Builds the graph Dcg for each rule of the grammar. Ex: Dcg E(1) : (1) E:= E + T. graph G: DP(1) U Down(E) U Down(T) Project(G,{attributes of E})

Definition at line 262 of file Builder_graphs.cpp.

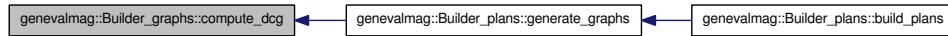
References attr_grammar, genevalmag::current_rule, genevalmag::Rule::get_left_symbol(), genevalmag::Rule::get_non_terminals_right_side(), genevalmag::Attr_grammar::get_rules(), genevalmag::Rule::key(), utilities::merge_graph(), p_Dcg_graphs, p_Down_graphs, p_Dp_graphs, and utilities::project_graph().

Referenced by genevalmag::Builder_plans::generate_graphs().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.7 bool genevalmag::Builder_graphs::compute_dependency_graphs ()

Algorithm DP Builds a graph dp for each rule of the grammar. Ex: E:= E + T graph: vertex: E,T Edges: E --> E T---> E

Returns

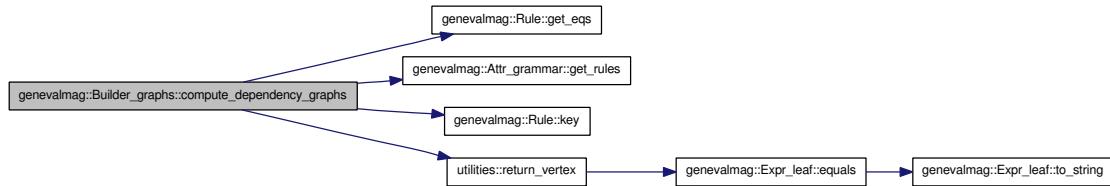
Algorithm DP Builds a graph dp for each rule of the grammar. Ex: E:= E + T graph: vertex: E,T Edges: E --> E T---> E

Definition at line 99 of file Builder_graphs.cpp.

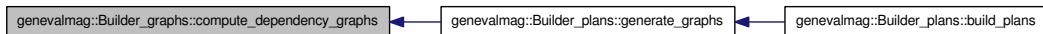
References attr_grammar, genevalmag::Rule::get_eqs(), genevalmag::Attr_grammar::get_rules(), genevalmag::Rule::key(), p_Dp_graphs, and utilities::return_vertex().

Referenced by genevalmag::Builder_plans::generate_graphs().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.8 bool genevalmag::Builder_graphs::compute_down_graph ()

Algorithm Down Builds the graphs down for each symbol of the grammar. Ex: Down(E) : (1) E:= E + T. graph G: DP(1) U Down(E) U Down(T) Project(G,{attributes of E})

Returns

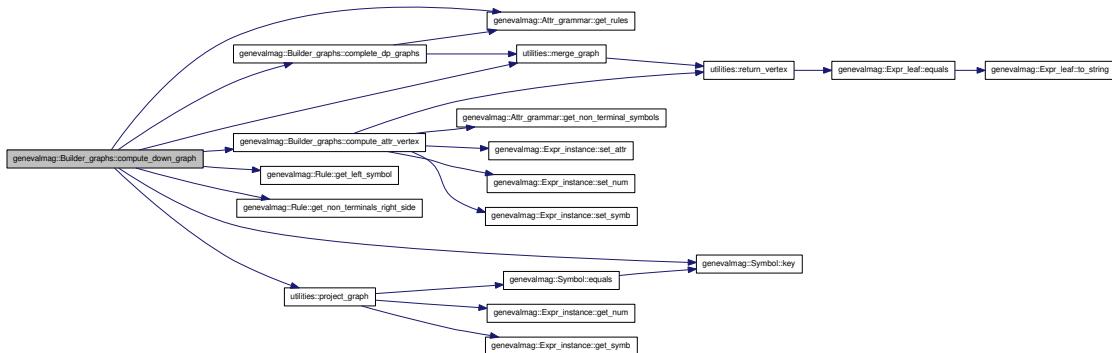
Algorithm Down Builds the graphs down for each symbol of the grammar. Ex: Down(E) : (1) E:= E + T. graph G: DP(1) U Down(E) U Down(T) Project(G,{attributes of E})

Definition at line 201 of file Builder_graphs.cpp.

References attr_grammar, attr_vertex_graphs, complete_dp_graphs(), compute_attr_vertex(), genevalmag::current_rule, genevalmag::Rule::get_left_symbol(), genevalmag::Rule::get_non_terminals_right_side(), genevalmag::Attr_grammar::get_rules(), genevalmag::Symbol::key(), utilities::merge_graph(), p_Down_graphs, p_Dp_graphs, and utilities::project_graph().

Referenced by genevalmag::Builder_plans::generate_graphs().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.9 const map< vector< unsigned short >, Graph > & genevalmag::Builder_graphs::get_adp_graphs() const

Returns the map with all ADP graphs creates.

Returns

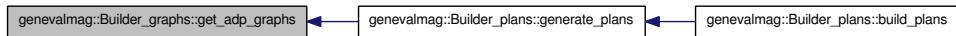
Returns the map with all ADP graphs creates.

Definition at line 70 of file Builder_graphs.cpp.

References p_Adp_graphs.

Referenced by genevalmag::Builder_plans::generate_plans().

Here is the caller graph for this function:



7.6.3.10 const map< vector< unsigned short >, Graph > & genevalmag::Builder_graphs::get_cyclic_graphs() const

Returns the map with all ADP with cycle graphs creates.

Returns

Returns the map with all ADP graphs creates.

Definition at line 78 of file Builder_graphs.cpp.

References p_Adp_subgraphs_cyclics.

7.6.3.11 const Graph & genevalmag::Builder_graphs::get_dcg_graph (unsigned short index_rule) const

Returns the down graph of the symbol.

Parameters

index_rule

Returns

Returns the down graph of the symbol.

Definition at line 86 of file Builder_graphs.cpp.

References p_Dcg_graphs.

Referenced by genevalmag::Builder_plans::generate_plans().

Here is the caller graph for this function:



7.6.3.12 bool genevalmag::Builder_graphs::save_adp_graphs (const string path_output) const

Saves all adp-graphs generates

Parameters

path_output

Returns

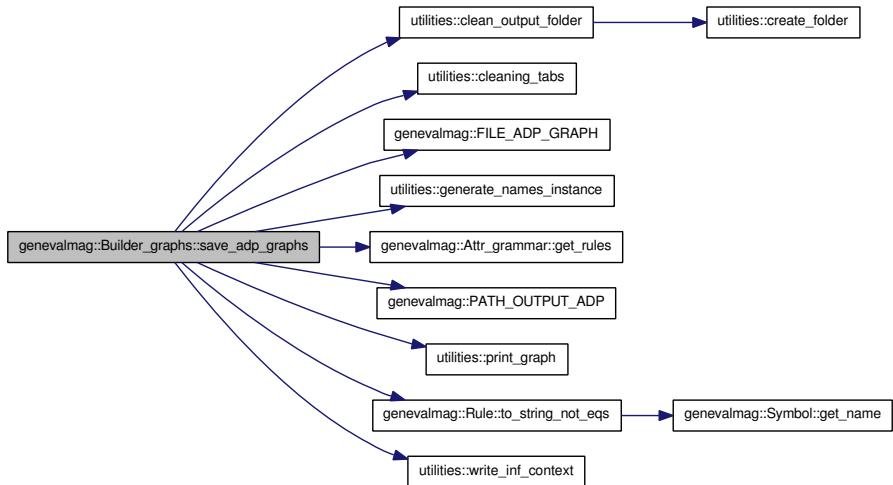
Saves all adp-graphs generates.

Definition at line 570 of file Builder_graphs.cpp.

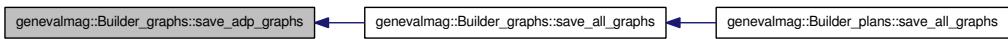
References attr_grammar, utilities::clean_output_folder(), utilities::cleaning_tabs(), genevalmag::FILE_ADP_GRAPH(), utilities::generate_names_instance(), genevalmag::Attr_grammar::get_rules(), p_Adp_graphs, genevalmag::PATH_OUTPUT_ADP(), utilities::print_graph(), genevalmag::Rule::to_string_not_eqs(), and utilities::write_inf_context().

Referenced by save_all_graphs().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.13 bool genevalmag::Builder_graphs::save_all_graphs (const string path_output) const

Saves all graphs generates: DP, Down, DCG and ADP.

Parameters

path_output

Returns

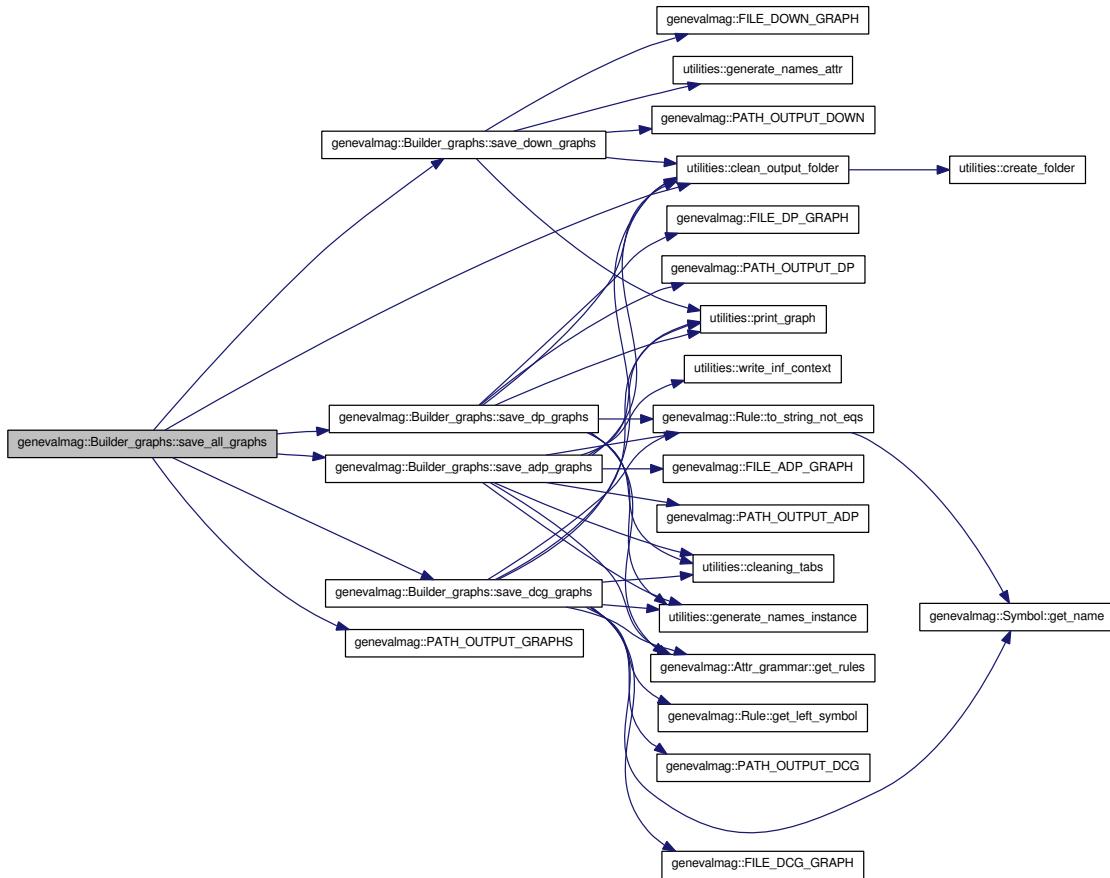
Saves all graphs generates: DP, Down, DCG and ADP.

Definition at line 602 of file Builder_graphs.cpp.

References utilities::clean_output_folder(), genevalmag::PATH_OUTPUT_GRAPHS(), save_adp_graphs(), save_dcg_graphs(), save_down_graphs(), and save_dp_graphs().

Referenced by genevalmag::Builder_plans::save_all_graphs().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.14 `bool genevalmag::Builder_graphs::save_cyclic_graphs (const string path_output) const`

Saves all graphs graph cycles.

Parameters

`path_output`

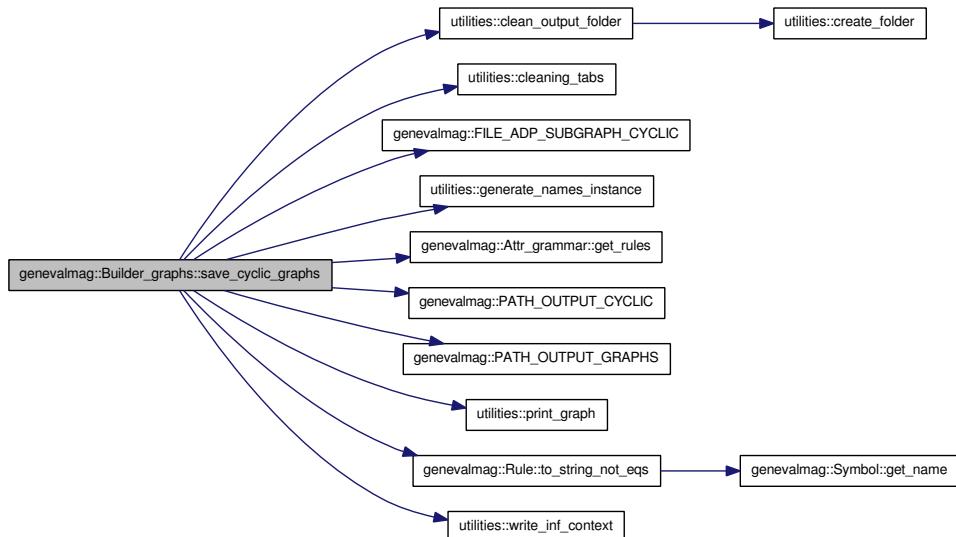
Returns

Definition at line 620 of file `Builder_graphs.cpp`.

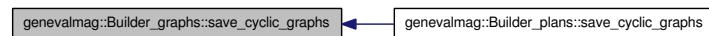
References attr_grammar, utilities::clean_output_folder(), utilities::cleaning_tabs(), genevalmag::FILE_AD_P_SUBGRAPH_CYCLIC(), utilities::generate_names_instance(), genevalmag::Attr_grammar::get_rules(), p_AdP_subgraphs_cyclics, genevalmag::PATH_OUTPUT_CYCLIC(), genevalmag::PATH_OUTPUT_GRAPHS(), utilities::print_graph(), genevalmag::Rule::to_string_not_eqs(), and utilities::write_inf_context().

Referenced by genevalmag::Builder_plans::save_cyclic_graphs().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.15 bool genevalmag::Builder_graphs::save_dcg_graphs (const string path_output) const

Saves all dcg-graphs generates

Parameters

path_output

Returns

Saves all dcg-graphs generates.

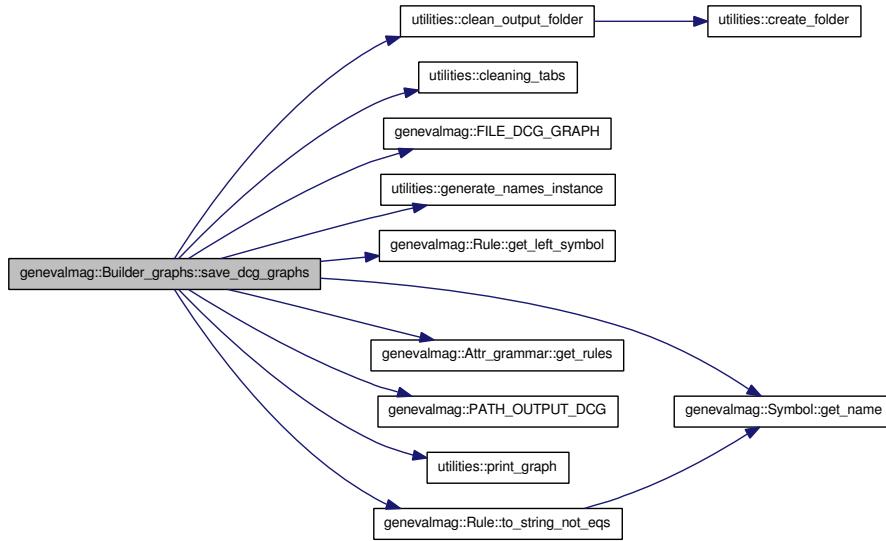
Definition at line 540 of file Builder_graphs.cpp.

References attr_grammar, utilities::clean_output_folder(), utilities::cleaning_tabs(), genevalmag::current_rule, genevalmag::FILE_DCG_GRAPH(), utilities::generate_names_instance(), genevalmag::Rule::get_left_symbol(), genevalmag::Symbol::get_name(),

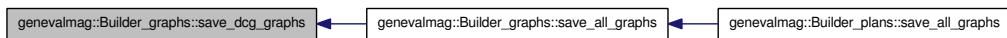
`genevalmag::Attr_grammar::get_rules()`, `p_Dcg_graphs`, `genevalmag::PATH_OUTPUT_DCG()`, `utilities::print_graph()`, and `genevalmag::Rule::to_string_not_eqs()`.

Referenced by `save_all_graphs()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.16 `bool genevalmag::Builder_graphs::save_down_graphs (const string path_output) const`

Saves all down-graphs generates

Parameters

`path_output`

Returns

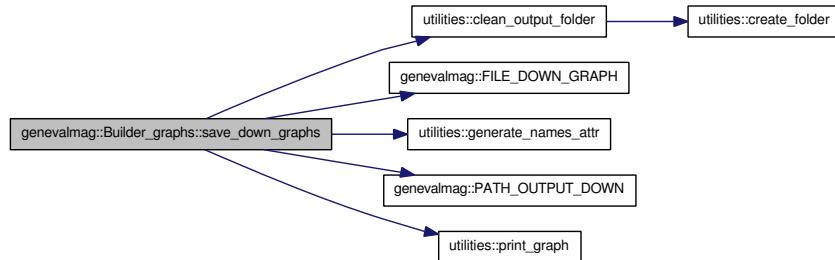
Saves all down-graphs generates.

Definition at line 514 of file `Builder_graphs.cpp`.

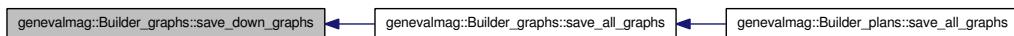
References `utilities::clean_output_folder()`, `genevalmag::FILE_DOWN_GRAPH()`, `utilities::generate_names_attr()`, `p_Down_graphs`, `genevalmag::PATH_OUTPUT_DOWN()`, and `utilities::print_graph()`.

Referenced by `save_all_graphs()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.3.17 `bool genevalmag::Builder_graphs::save_dp_graphs (const string path_output) const`

Saves all dp-graphs generates

Parameters

path_output

Returns

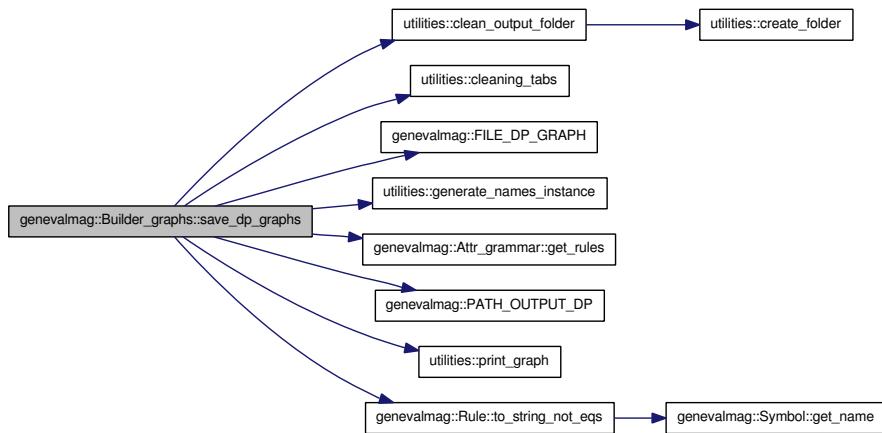
Saves all dp-graphs generates

Definition at line 486 of file Builder_graphs.cpp.

References attr_grammar, utilities::clean_output_folder(), utilities::cleaning_tabs(), genevalmag::current_rule, genevalmag::FILE_DP_GRAPH(), utilities::generate_names_instance(), genevalmag::Attr_grammar::get_rules(), p_Dp_graphs, genevalmag::PATH_OUTPUT_DP(), utilities::print_graph(), and genevalmag::Rule::to_string_not_eqs().

Referenced by save_all_graphs().

Here is the call graph for this function:



Here is the caller graph for this function:



7.6.4 Member Data Documentation

7.6.4.1 genevalmag::Builder_graphs::attr_grammar [private]

References to the attribute grammar.

Definition at line 47 of file Builder_graphs.h.

Referenced by complete_dp_graphs(), compute_adp_graph(), compute_attr_vertex(), compute_dcg(), compute_dependency_graphs(), compute_down_graph(), save_adp_graphs(), save_cyclic_graphs(), save_dcg_graphs(), and save_dp_graphs().

7.6.4.2 genevalmag::Builder_graphs::attr_vertex_graphs [private]

Store the vertex-attr graphs. The key corresponds to the key [Symbol](#).

Definition at line 58 of file Builder_graphs.h.

Referenced by complete_dp_graphs(), compute_attr_vertex(), and compute_down_graph().

7.6.4.3 genevalmag::Builder_graphs::p_AdP_graphs [private]

Store the adp graphs. The key corresponds to the key [Rule](#) with the inferior context.

Definition at line 74 of file Builder_graphs.h.

Referenced by check_cyclic_adp_dependencies(), combined_inf_contexts(), get_adp_graphs(), and save_adp_graphs().

7.6.4.4 genevalmag::Builder_graphs::p_AdP_subgraphs_cyclics [private]

Store the adp graphs. The key corresponds to the key [Rule](#) with the inferior context.

Definition at line 79 of file [Builder_graphs.h](#).

Referenced by [check_cyclic_adp_dependencies\(\)](#), [get_cyclic_graphs\(\)](#), and [save_cyclic_graphs\(\)](#).

7.6.4.5 genevalmag::Builder_graphs::p_Dcg_graphs [private]

Store the dcg graphs. The key corresponds to the key [Rule](#).

Definition at line 69 of file [Builder_graphs.h](#).

Referenced by [combined_inf_contexts\(\)](#), [compute_deg\(\)](#), [get_dcg_graph\(\)](#), and [save_deg_graphs\(\)](#).

7.6.4.6 genevalmag::Builder_graphs::p_Down_graphs [private]

Store the down graphs. The key corresponds to the key [Symbol](#).

Definition at line 63 of file [Builder_graphs.h](#).

Referenced by [compute_dcg\(\)](#), [compute_down_graph\(\)](#), and [save_down_graphs\(\)](#).

7.6.4.7 genevalmag::Builder_graphs::p_Dp_graphs [private]

Store the DP graphs. The key corresponds to the key [Rule..](#)

Definition at line 53 of file [Builder_graphs.h](#).

Referenced by [complete_dp_graphs\(\)](#), [compute_adp_graph\(\)](#), [compute_attr_vertex\(\)](#), [compute_dcg\(\)](#), [compute_dependency_graphs\(\)](#), [compute_down_graph\(\)](#), and [save_dp_graphs\(\)](#).

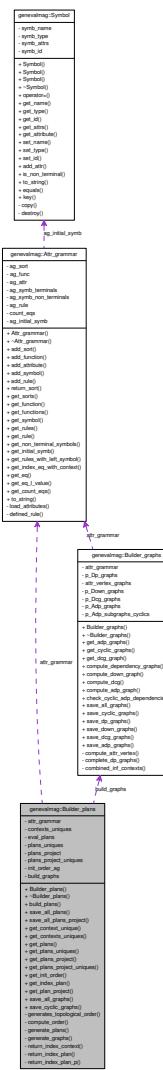
The documentation for this class was generated from the following files:

- [include/Builders/Builder_graphs.h](#)
- [src/Builders/Builder_graphs.cpp](#)

7.7 genevalmag::Builder_plans Class Reference

```
#include <Builder_plans.h>
```

Collaboration diagram for genevalmag::Builder_plans:



Public Member Functions

- `Builder_plans` (`const Attr_grammar &attribute_grammar`)
 - `~Builder_plans` ()
 - `unsigned short build_plans` ()
 - `bool save_all_plans` (`const string path_output`) const
 - `bool save_all_plans_project` (`const string path_output`) const
 - `const Order_rule & get_context_unique` (`const unsigned short i_context`) const
 - `const vector< Order_rule > & get_contexts_uniques` () const
 - `const map< Key_plan, unsigned short > & get_plans` () const
 - `const vector< Order_eval_eq > & get_plans_uniques` () const

- const map< Key_plan_project, unsigned short > & get_plans_project () const
- const vector< Order_eval_eq > & get_plans_project_uniques () const
- const unsigned short & get_init_order () const
- const unsigned short get_index_plan (const map< Key_plan, unsigned short >::const_iterator it_plan) const
- const map< Key_plan_project, unsigned short >::const_iterator get_plan_project (const Key_plan_project &key) const
- void save_all_graphs (const string path_output) const
- void save_cyclic_graphs (const string path_output) const

Private Member Functions

- void generates_topological_order (const Graph &graph, Order_eval_eq &result_order, const Context_rule &context_rule) const
- Order_eval_eq compute_order (const Graph &graph_adp, unsigned short index_order, const Context_rule &context_rule)
- bool generate_plans ()
- bool generate_graphs ()
- unsigned short return_index_context (const Order_rule &order)
- unsigned short return_index_plan (const Order_eval_eq &order)
- unsigned short return_index_plan_p (const Order_eval_eq &order)

Private Attributes

- const Attr_grammar & attr_grammar
References to the attribute grammar.
- vector< Order_rule > contexts_uniques
Store all different context rule generates in the grammar.
- map< Key_plan, unsigned short > eval_plans
Store all evaluation plans of the grammar. Saves index of plans_uniques vector. "ro" function. Wuu Yang's paper.
- vector< Order_eval_eq > plans_uniques
- map< Key_plan_project, unsigned short > plans_project
Store all evaluation plans projects of the grammar. Saves index of plans_project_uniques vector. "theta" function. Wuu Yang's paper.
- vector< Order_eval_eq > plans_project_uniques
- unsigned short init_order_ag
Store the initial evaluation order of the attributes of the initial symbol.
- Builder_graphs build_graphs
Stores all graphs DP, DOWN, DCG and ADP, generated for the grammar's analysys.

7.7.1 Detailed Description

Definition at line 254 of file Builder_plans.h.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 `genevalmag::Builder_plans::Builder_plans (const Attr_grammar & attribute_grammar)`

Constructor empty of Builder plans.

Parameters

attribute_grammar

Returns

Constructor empty of Builder plan.

Definition at line 34 of file Builder_plans.cpp.

7.7.2.2 `genevalmag::Builder_plans::~Builder_plans ()`

Destructor of Builder plans.

Returns

Destructor of Builder plan.

Definition at line 43 of file Builder_plans.cpp.

7.7.3 Member Function Documentation

7.7.3.1 `unsigned short genevalmag::Builder_plans::build_plans ()`

Generates and saves all evaluation's plans for the [Attribute](#) Grammar.

Return: 0: when all success process. 1: when detects cilmcy graph. 2: when detects error in the graph generation.

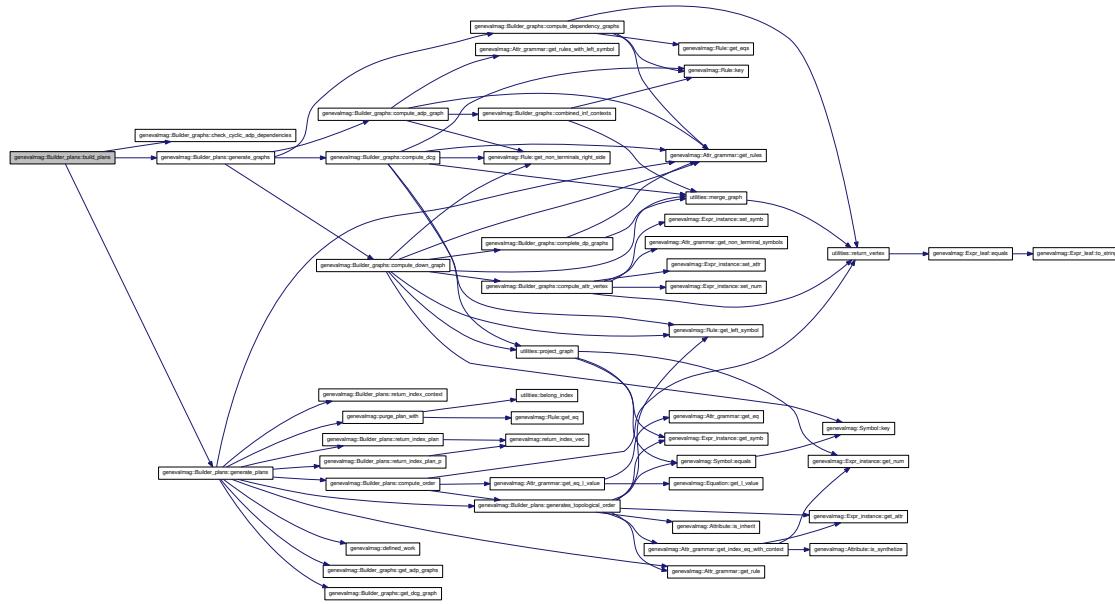
Returns

Return: 0: when all success process. 1: when detects cilmcy graph. 2: when detects error in the graph generation.

Definition at line 521 of file Builder_plans.cpp.

References `build_graphs`, `genevalmag::Builder_graphs::check_cyclic_adp_dependencies()`, `generate_graphs()`, and `generate_plans()`.

Here is the call graph for this function:



7.7.3.2 Order_eval_eq genevalmag::Builder_plans::compute_order (const Graph & graph_adp, unsigned short index_order, const Context_rule & context_rule) [private]

Compute the rule's order. The changes are applies about paramenter "result_order".

Parameters

graph_adp

index_order

context_rule

Returns

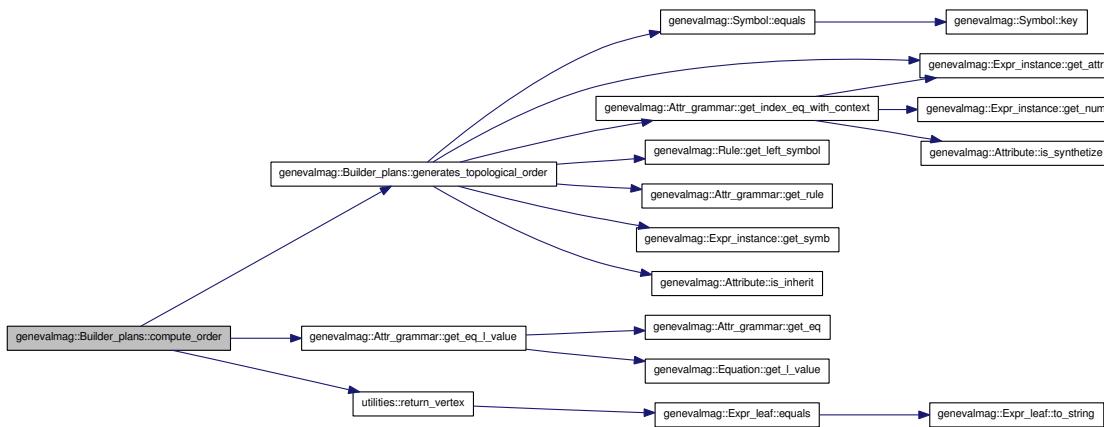
Compute the rule's order. The changes are applies about paramenter "result_order".

Definition at line 134 of file Builder_plans.cpp.

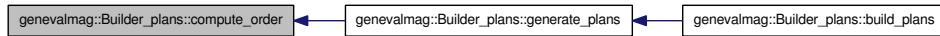
References attr_grammar, generates_topological_order(), genevalmag::Attr_grammar::get_eq_l_value(), plans_project_uniques, and utilities::return_vertex().

Referenced by generate_plans().

Here is the call graph for this function:



Here is the caller graph for this function:



7.7.3.3 `bool genevalmag::Builder_plans::generate_graphs()` [private]

Generates all graphs for attribute grammar: DP, DOWN, DCG and ADP.

Returns

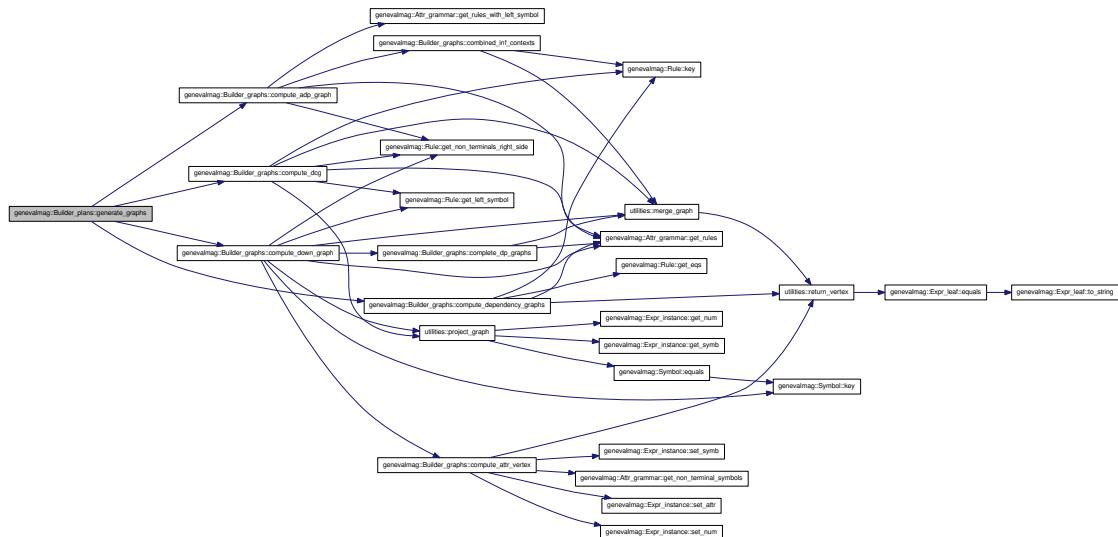
Generates all graphs for attribute grammar: DP, DOWN, DCG and ADP.

Definition at line 66 of file Builder_plans.cpp.

References build_graphs, genevalmag::Builder_graphs::compute_adp_graph(), genevalmag::Builder_graphs::compute_dcg(), genevalmag::Builder_graphs::compute_dependency_graphs(), and genevalmag::Builder_graphs::compute_down_graph().

Referenced by build_plans().

Here is the call graph for this function:



Here is the caller graph for this function:



7.7.3.4 bool genevalmag::Builder_plans::generate_plans () [private]

Generates and saves all evaluation's plans for the [Attribute](#) Grammar.

Returns

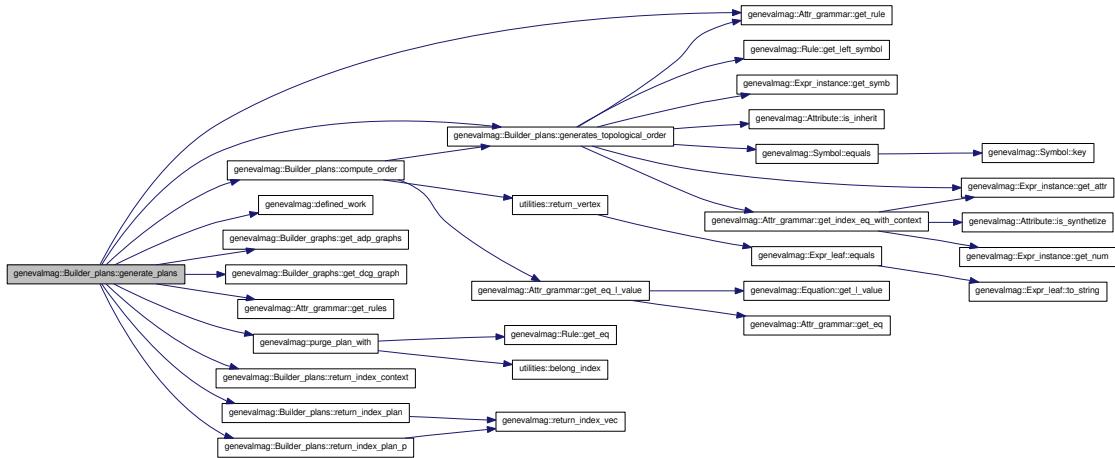
Generates and saves all evaluation's plans for the [Attribute](#) Grammar.

Definition at line 403 of file Builder_plans.cpp.

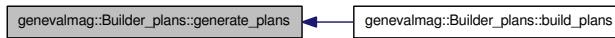
References attr_grammar, build_graphs, compute_order(), genevalmag::c_rule::context, genevalmag::defined_work(), eval_plans, genevalmag::k_w::father, genevalmag::c_rule::father, generates_topological_order(), genevalmag::Builder_graphs::get_adp_graphs(), genevalmag::Builder_graphs::get_dcg_graph(), genevalmag::Attr_grammar::get_rule(), genevalmag::Attr_grammar::get_rules(), genevalmag::k_plan::id_plan, genevalmag::k_p_project::id_plan_project, genevalmag::k_w::id_rule, genevalmag::k_p_project::index_occurrence, init_order_ag, genevalmag::i_w::item, genevalmag::i_w::order_attr, genevalmag::k_plan::plan, plans_project, genevalmag::purge_plan_with(), return_index_context(), return_index_plan(), return_index_plan_p(), and genevalmag::k_p_project::symbol_project.

Referenced by build_plans().

Here is the call graph for this function:



Here is the caller graph for this function:



7.7.3.5 void genevalmag::Builder_plans::generates_topological_order (const Graph & graph, Order_eval_eq & result_order, const Context_rule & context_rule) const [private]

Applies a topological order at graph. The changes are applies about parameter "result_order".

Parameters

graph

result_order

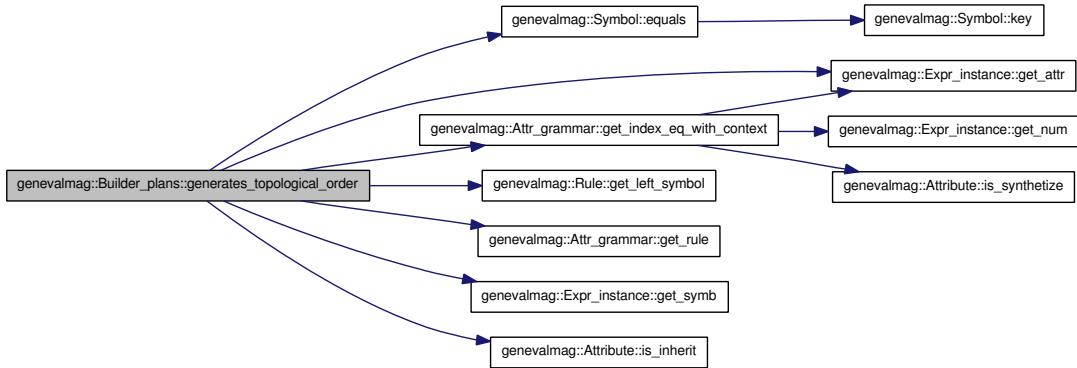
context_rule Applies a topological order at graph. The changes are applies about parameter "result_order".

Definition at line 90 of file Builder_plans.cpp.

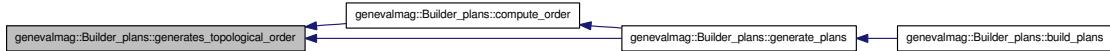
References attr_grammar, genevalmag::c_rule::context, genevalmag::Symbol::equals(), genevalmag::c_rule::father, genevalmag::Expr_instance::get_attr(), genevalmag::Attr_grammar::get_index_eq_with_context(), genevalmag::Rule::get_left_symbol(), genevalmag::Attr_grammar::get_rule(), genevalmag::Expr_instance::get_symb(), and genevalmag::Attribute::is_inherit().

Referenced by compute_order(), and generate_plans().

Here is the call graph for this function:



Here is the caller graph for this function:



7.7.3.6 const Order_rule & genevalmag::Builder_plans::get_context_unique (const unsigned short i_context) const

Returns the context rule unique in this index.

Parameters

i_context

Returns

Returns the context rule unique in this index.

Definition at line 543 of file Builder_plans.cpp.

References contexts_uniques.

Referenced by genevalmag::Builder_visit_sequences::generate_visit_sequences().

Here is the caller graph for this function:



7.7.3.7 const vector< Order_rule > & genevalmag::Builder_plans::get_contexts_uniques () const

Returns all contexts rule uniques.

Returns

Returns all contexts rule uniques.

Definition at line 551 of file Builder_plans.cpp.

References contexts_uniques.

Referenced by genevalmag::Builder_code::generate_initialize_context().

Here is the caller graph for this function:



7.7.3.8 const unsigned short genevalmag::Builder_plans::get_index_plan (const map<Key_plan, unsigned short >::const_iterator it_plan) const

Returns the index inside the map of the plan that passed as parameter.

Parameters

it_plan

Returns

Returns the index inside the map of the plan that passed as parameter.

Definition at line 599 of file Builder_plans.cpp.

References eval_plans.

7.7.3.9 const unsigned short & genevalmag::Builder_plans::get_init_order () const

Returns the intial order of attributes of the initial symbol.

Returns

Returns the intial order of attributes of the initial symbol.

Definition at line 591 of file Builder_plans.cpp.

References init_order_ag.

Referenced by genevalmag::Builder_code::generate_evaluator().

Here is the caller graph for this function:



7.7.3.10 `const map< Key_plan_project, unsigned short >::const_iterator genevalmag::Builder_plans::get_plan_project (const Key_plan_project & key) const`

Returns an constant iterator of a project plan with key equals at the key that passed as parameter. Always finds the element.

Parameters

`key`

Returns

Returns an constant iterator of a project plan with key equals at the key that passed as parameter. Always finds the element.

Definition at line 618 of file Builder_plans.cpp.

References plans_project.

7.7.3.11 `const map< Key_plan, unsigned short > & genevalmag::Builder_plans::get_plans () const`

Returns all evaluations plans.

Returns

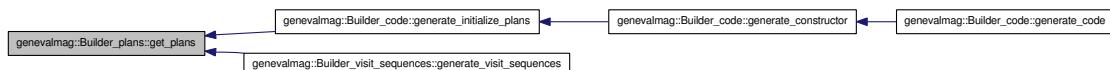
Returns all evaluations plans.

Definition at line 559 of file Builder_plans.cpp.

References eval_plans.

Referenced by genevalmag::Builder_code::generate_initialize_plans(), and genevalmag::Builder_visit_sequences::generate_visit_sequences().

Here is the caller graph for this function:



7.7.3.12 `const map< Key_plan_project, unsigned short > & genevalmag::Builder_plans::get_plans_project () const`

Returns all evaluations plans project.

Returns

Returns all evaluations plans project.

Definition at line 575 of file Builder_plans.cpp.

References plans_project.

Referenced by genevalmag::Builder_code::generate_initialize_plan_proj().

Here is the caller graph for this function:



7.7.3.13 const vector< Order_eval_eq > & genevalmag::Builder_plans::get_plans_project_uniques () const

Returns all evaluations plans project uniques.

Returns

Returns all evaluations plans project uniques.

Definition at line 583 of file Builder_plans.cpp.

References plans_project_uniques.

7.7.3.14 const vector< Order_eval_eq > & genevalmag::Builder_plans::get_plans_uniques () const

Returns all evaluations plans uniques.

Returns

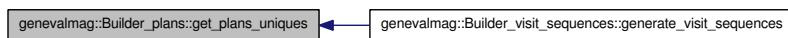
Returns all evaluations plans uniques.

Definition at line 567 of file Builder_plans.cpp.

References plans_uniques.

Referenced by genevalmag::Builder_visit_sequences::generate_visit_sequences().

Here is the caller graph for this function:



7.7.3.15 unsigned short genevalmag::Builder_plans::return_index_context (const Order_rule & order) [private]

Returns the index in the vector of context, or inserts in the last position.

Parameters*order***Returns**

Returns the index in the vector of context, or inserts in the last position.

Definition at line 353 of file Builder_plans.cpp.

References contexts_uniques.

Referenced by generate_plans().

Here is the caller graph for this function:



7.7.3.16 **unsigned short genevalmag::Builder_plans::return_index_plan (const Order_eval_eq & order) [private]**

Returns the index in the vector of plans, or inserts in the last position.

Parameters*order***Returns**

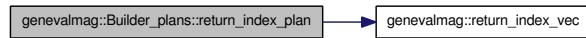
Returns the index in the vector of plans, or inserts in the last position.

Definition at line 387 of file Builder_plans.cpp.

References plans_uniques, and genevalmag::return_index_vec().

Referenced by generate_plans().

Here is the call graph for this function:



Here is the caller graph for this function:



7.7.3.17 **unsigned short genevalmag::Builder_plans::return_index_plan_p (const Order_eval_eq & order) [private]**

Returns the index in the vector of plans projects, or inserts in the last position.

Parameters

order

Returns

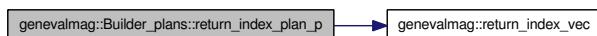
Returns the index in the vector of plans projects, or inserts in the last position.

Definition at line 395 of file Builder_plans.cpp.

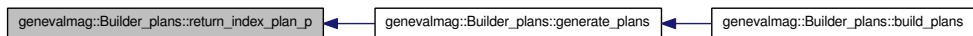
References plans_project_uniques, and genevalmag::return_index_vec().

Referenced by generate_plans().

Here is the call graph for this function:



Here is the caller graph for this function:



7.7.3.18 void genevalmag::Builder_plans::save_all_graphs (const string path_output) const

Saves all graphs generated as the analysis of the dependencies between attributes.

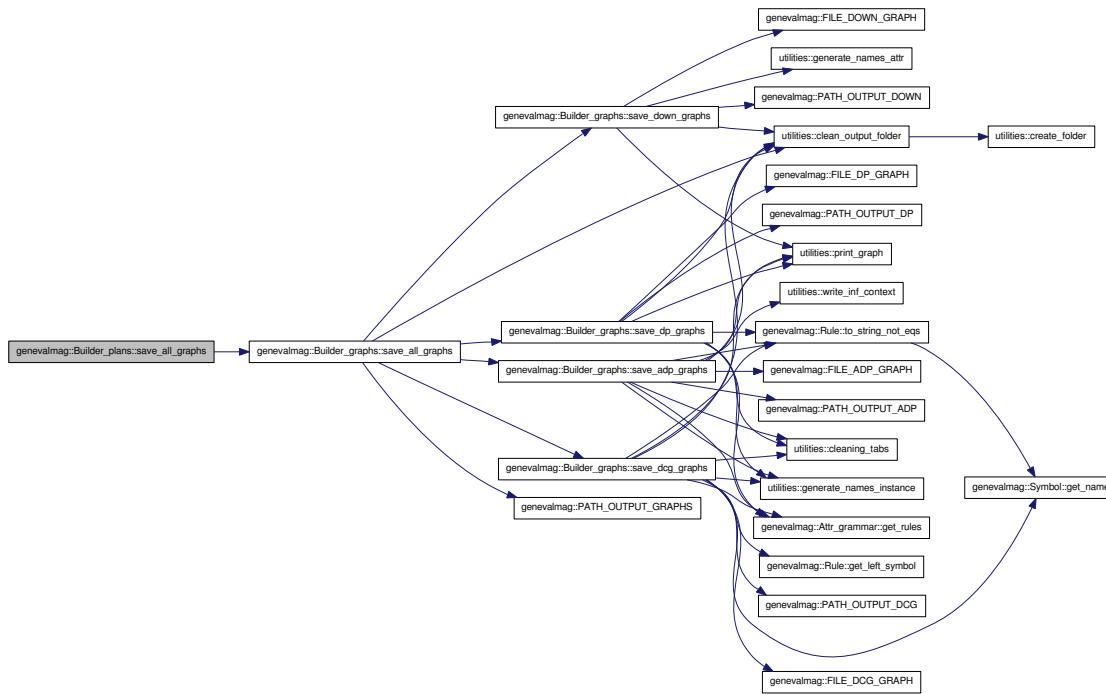
Parameters

path_output Saves all graphs generated as the analysis of the dependencies between attributes.

Definition at line 50 of file Builder_plans.cpp.

References build_graphs, and genevalmag::Builder_graphs::save_all_graphs().

Here is the call graph for this function:



7.7.3.19 `bool genevalmag::Builder_plans::save_all_plans (const string path_output) const`

Saves all plans. Creates a graph that represents the plan and uses print_graph with dot.

Parameters

path_output

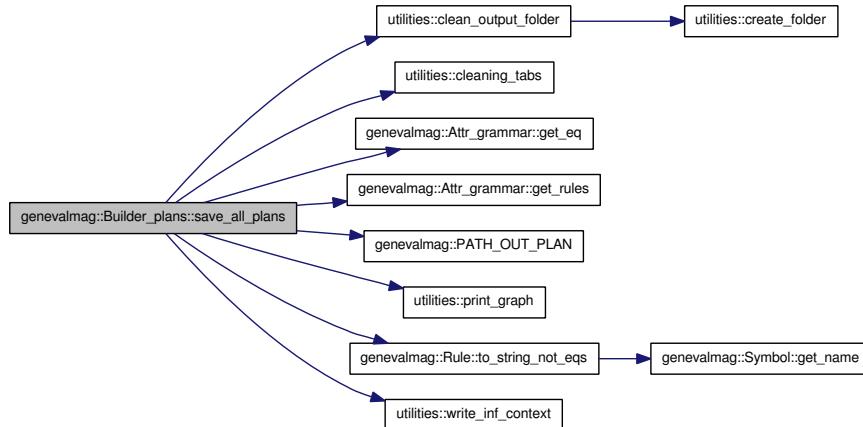
Returns

Saves all plans. Creates a graph that represents the plan and uses print_graph with dot.

Definition at line 171 of file Builder_plans.cpp.

References attr_grammar, utilities::clean_output_folder(), utilities::cleaning_tabs(), contexts_uniques, eval_plans, genevalmag::Attr_grammar::get_eq(), genevalmag::Attr_grammar::get_rules(), genevalmag::PATH_OUT_PLAN(), plans_project_uniques, plans_uniques, utilities::print_graph(), genevalmag::Rule::to_string_not_eqs(), and utilities::write_inf_context().

Here is the call graph for this function:



7.7.3.20 **bool genevalmag::Builder_plans::save_all_plans_project (const string path_output) const**

Saves all projected's plans. Creates a graph that represents the plan and uses print_graph with dot.

Parameters

path_output

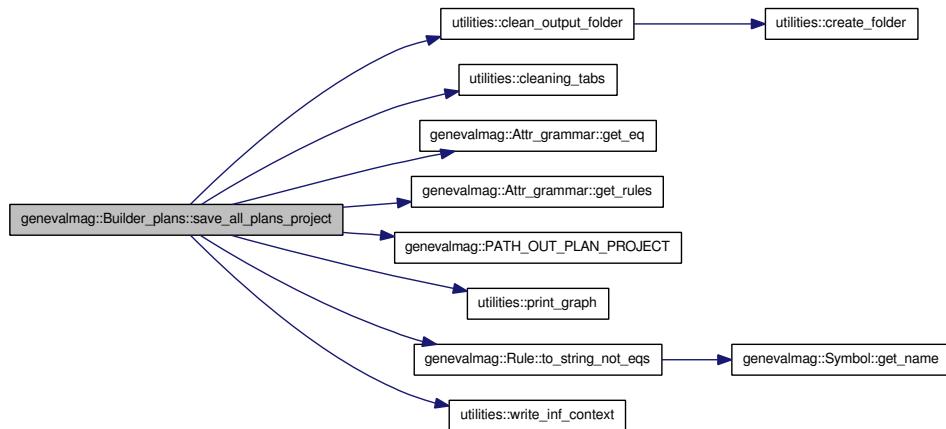
Returns

Saves all projected's plans. Creates a graph that represents the plan and uses print_graph with dot.

Definition at line 242 of file Builder_plans.cpp.

References attr_grammar, utilities::clean_output_folder(), utilities::cleaning_tabs(), contexts_uniques, genevalmag::Attr_grammar::get_eq(), genevalmag::Attr_grammar::get_rules(), genevalmag::PATH_OUT_PLAN_PROJECT(), plans_project, plans_project_uniques, utilities::print_graph(), genevalmag::Rule::to_string_not_eqs(), and utilities::write_inf_context().

Here is the call graph for this function:



7.7.3.21 void genevalmag::Builder_plans::save_cyclic_graphs (const string path_output) const

Saves the graphs generated as the analysis of the dependencies between attributes, which demonstrate cyclicity.

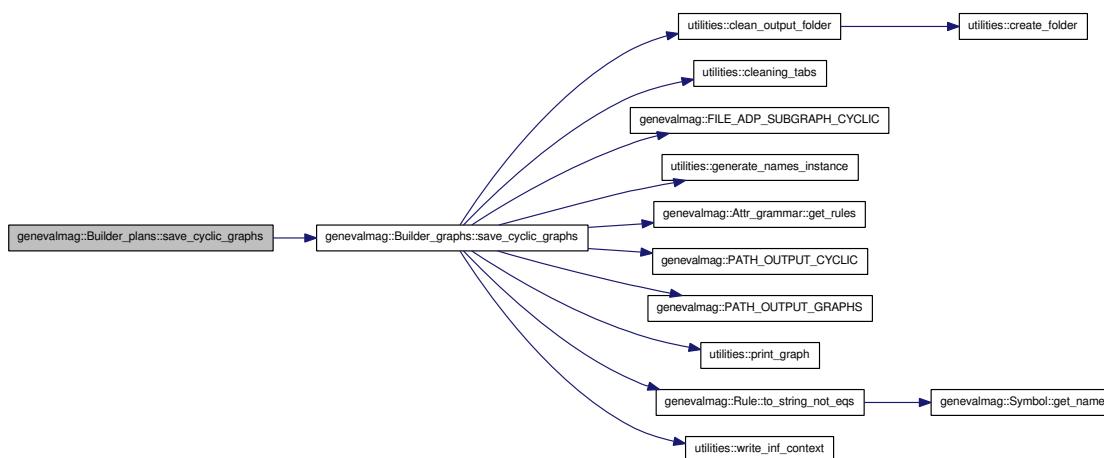
Parameters

path_output Saves the graphs generated as the analysis of the dependencies between attributes, which demonstrate cyclicity.

Definition at line 58 of file Builder_plans.cpp.

References build_graphs, and genevalmag::Builder_graphs::save_cyclic_graphs().

Here is the call graph for this function:



7.7.4 Member Data Documentation

7.7.4.1 genevalmag::Builder_plans::attr_grammar [private]

References to the attribute grammar.

Definition at line 261 of file Builder_plans.h.

Referenced by compute_order(), generate_plans(), generates_topological_order(), save_all_plans(), and save_all_plans_project().

7.7.4.2 genevalmag::Builder_plans::build_graphs [private]

Stores all graphs DP, DOWN, DCG and ADP, generated for the grammar's analysis.

Definition at line 295 of file Builder_plans.h.

Referenced by build_plans(), generate_graphs(), generate_plans(), save_all_graphs(), and save_cyclic_graphs().

7.7.4.3 genevalmag::Builder_plans::contexts_uniques [private]

Store all different context rule generates in the grammar.

Definition at line 267 of file Builder_plans.h.

Referenced by get_context_unique(), get_contexts_uniques(), return_index_context(), save_all_plans(), and save_all_plans_project().

7.7.4.4 genevalmag::Builder_plans::eval_plans [private]

Store all evaluation plans of the grammar. Saves index of plans_uniques vector. "ro" function. Wuu Yang's paper.

Definition at line 274 of file Builder_plans.h.

Referenced by generate_plans(), get_index_plan(), get_plans(), and save_all_plans().

7.7.4.5 genevalmag::Builder_plans::init_order_ag [private]

Store the initial evaluation order of the attributes of the initial symbol.

Definition at line 289 of file Builder_plans.h.

Referenced by generate_plans(), and get_init_order().

7.7.4.6 genevalmag::Builder_plans::plans_project [private]

Store all evaluation plans projects of the grammar. Saves index of plans_project_uniques vector. "theta" function. Wuu Yang's paper.

Definition at line 282 of file Builder_plans.h.

Referenced by generate_plans(), get_plan_project(), get_plans_project(), and save_all_plans_project().

7.7.4.7 `vector< Order_eval_eq > genevalmag::Builder_plans::plans_project_uniques`
[private]

Definition at line 283 of file Builder_plans.h.

Referenced by compute_order(), get_plans_project_uniques(), return_index_plan_p(), save_all_plans(), and save_all_plans_project().

7.7.4.8 `vector< Order_eval_eq > genevalmag::Builder_plans::plans_uniques`
[private]

Definition at line 275 of file Builder_plans.h.

Referenced by get_plans_uniques(), return_index_plan(), and save_all_plans().

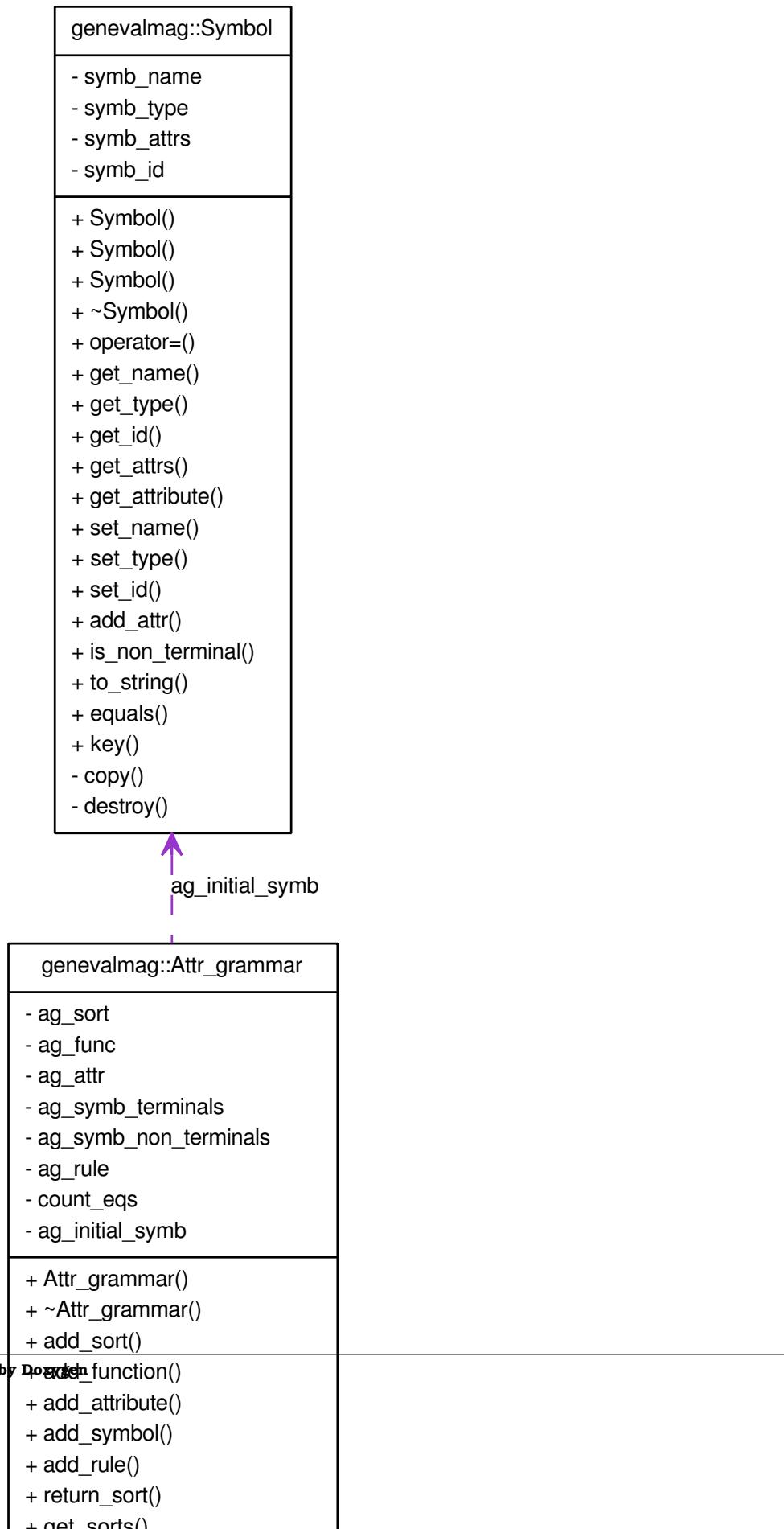
The documentation for this class was generated from the following files:

- include/Builders/Builder_plans.h
- src/Builders/Builder_plans.cpp

7.8 genevalmag::Builder_visit_sequences Class Reference

```
#include <Builder_visit_sequences.h>
```

Collaboration diagram for genevalmag::Builder_visit_sequences:



Public Member Functions

- `Builder_visit_sequences (const Builder_plans &builder_plan, const Attr_grammar &attribute_grammar)`
- `virtual ~Builder_visit_sequences ()`
- `bool generate_visit_sequences ()`
- `const vector< Visit_seq > & get_visit_seq () const`
- `void print_all_visit_sequences () const`

Private Member Functions

- `bool gen_visit_seq (const map< Key_plan, unsigned short >::const_iterator &it_plan, vector< Expr_instance > &ins_computed, vector< map< Key_plan, unsigned short >::const_iterator > &plans_computed, const vector< unsigned short > &v_seq_computed)`
- `void update_visit_sequence (const Visit_seq &sequence, const size_t i_plan)`

Private Attributes

- `const Builder_plans & b_plans`
References to all plans generates.
- `const Attr_grammar & attr_grammar`
References to the attribute grammar.
- `vector< Visit_seq > all_visit_seqs`
Stores all visit sequences generates.

7.8.1 Detailed Description

Definition at line 30 of file `Builder_visit_sequences.h`.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 `genevalmag::Builder_visit_sequences::Builder_visit_sequences (const Builder_plans & builder_plan, const Attr_grammar & attribute_grammar)`

Constructor empty of `Builder_visit_sequences`.

Parameters

builder_plan
attribute_grammar

Returns

Constructor empty of `Builder_visit_sequences`.

Definition at line 26 of file `Builder_visit_sequences.cpp`.

7.8.2.2 genevalmag::Builder_visit_sequences::~Builder_visit_sequences () [virtual]

Destructor of `Builder_visit_sequences`.

Returns

Destructor of `Builder_visit_sequences`.

Definition at line 38 of file `Builder_visit_sequences.cpp`.

7.8.3 Member Function Documentation**7.8.3.1 bool genevalmag::Builder_visit_sequences::gen_visit_seq (const map<Key_plan, unsigned short >::const_iterator & it_plan, vector<Expr_instance> & ins_computed, vector<map<Key_plan, unsigned short >::const_iterator> & plans_computed, const vector<unsigned short > & v_seq_computed) [private]**

Generates recursively the visit sequence, navigating the tree as indicated by the plans, visiting the children, returning to the parent or generating a compute attribute.

Parameters

it_plan

ins_computed

plans_computed

v_seq_computed

Returns

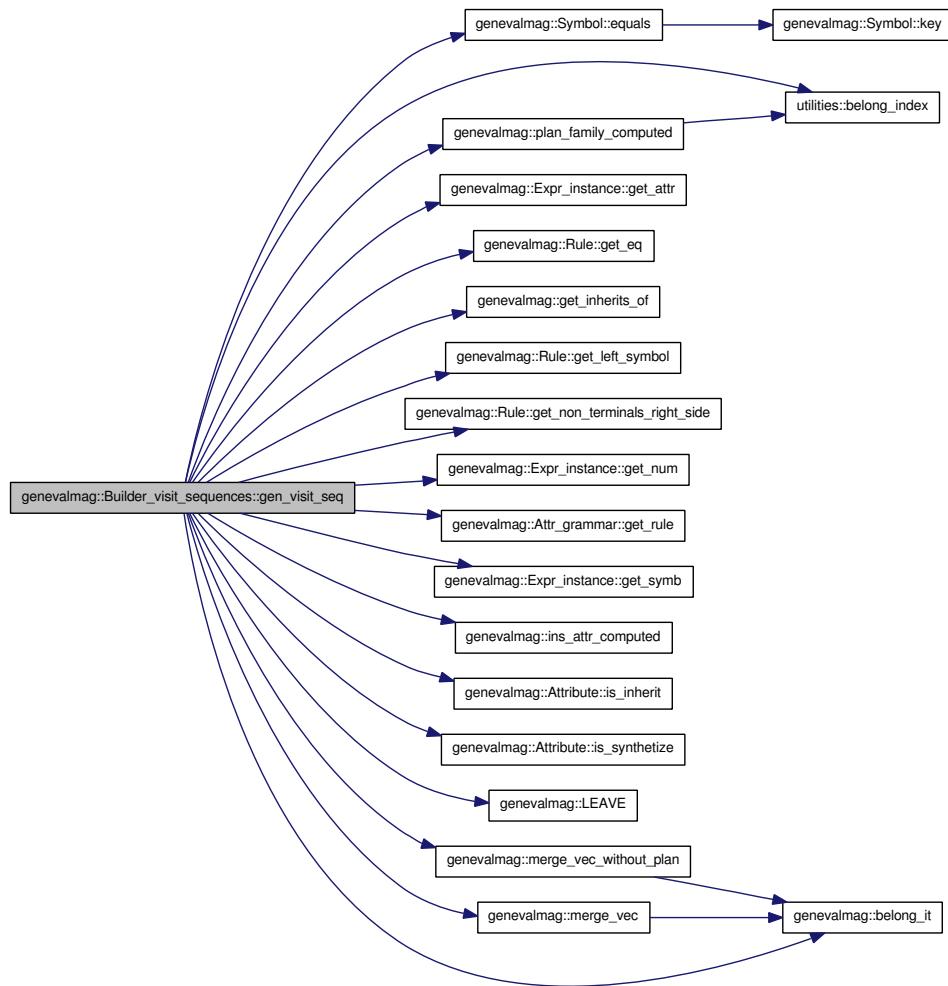
Generates recursively the visit sequence, navigating the tree as indicated by the plans, visiting the children, returning to the parent or generating a compute attribute.

Definition at line 162 of file `Builder_visit_sequences.cpp`.

References `genevalmag::attr_grammar`, `utilities::belong_index()`, `genevalmag::belong_it()`, `genevalmag::Symbol::equals()`, `genevalmag::Expr_instance::get_attr()`, `genevalmag::Rule::get_eq()`, `genevalmag::get_inherits_of()`, `genevalmag::Rule::get_left_symbol()`, `genevalmag::Rule::get_non_terminals_right_side()`, `genevalmag::Expr_instance::get_num()`, `genevalmag::Attr_grammar::get_rule()`, `genevalmag::Expr_instance::get_symb()`, `genevalmag::k_p_project::id_plan_project`, `genevalmag::k_p_project::index_occurrence`, `genevalmag::ins_attr_computed()`, `genevalmag::Attribute::is_inherit()`, `genevalmag::Attribute::is_synthesize()`, `genevalmag::LEAVE()`, `genevalmag::merge_vec()`, `genevalmag::merge_vec_without_plan()`, `genevalmag::plan_family_computed()`, and `genevalmag::k_p_project::symbol_project`.

Referenced by `generate_visit_sequences()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.8.3.2 `bool genevalmag::Builder_visit_sequences::generate_visit_sequences ()`

Generates a visit sequence for each evaluation plan.

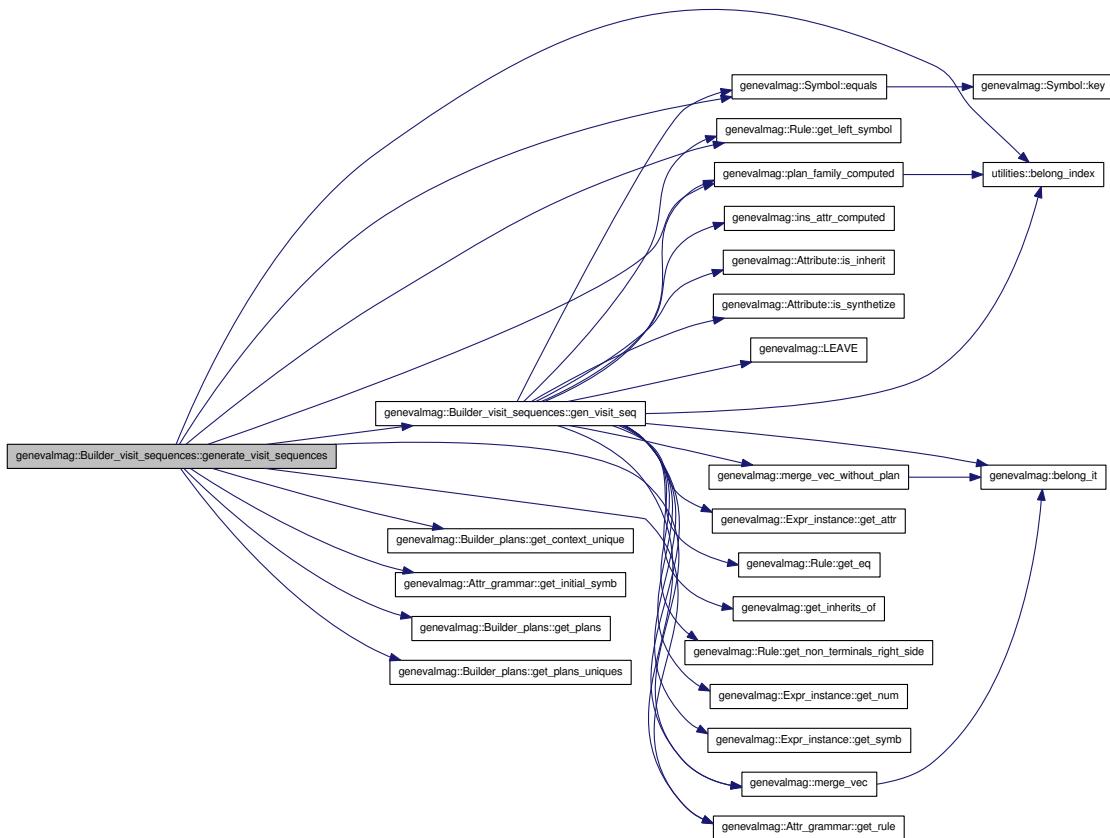
Returns

Generates a visit sequence for each evaluation plan.

Definition at line 358 of file `Builder_visit_sequences.cpp`.

References all_visit_seqs, attr_grammar, b_plans, utilities::belong_index(), genevalmag::Symbol::equals(), gen_visit_seq(), genevalmag::Builder_plans::get_context_unique(), genevalmag::Attr_grammar::get_initial_symb(), genevalmag::Rule::get_left_symbol(), genevalmag::Builder_plans::get_plans(), genevalmag::Builder_plans::get_plans_uniques(), genevalmag::Attr_grammar::get_rule(), genevalmag::merge_vec(), and genevalmag::plan_family_computed().

Here is the call graph for this function:



7.8.3.3 const vector< Visit_seq > & genevalmag::Builder_visit_sequences::get_visit_seq() const

Returns the vector with all visit sequences generates.

Returns

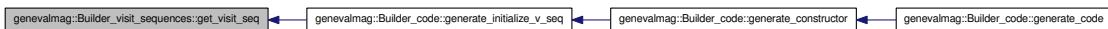
Returns the vector with all visit sequences generates.

Definition at line 393 of file Builder_visit_sequences.cpp.

References all_visit_seqs.

Referenced by genevalmag::Builder_code::generate_initialize_v_seq().

Here is the caller graph for this function:



7.8.3.4 void genevalmag::Builder_visit_sequences::print_all_visit_sequences () const

Print in console, all visit sequences generates.

Definition at line 401 of file Builder_visit_sequences.cpp.

References all_visit_seqs.

7.8.3.5 void genevalmag::Builder_visit_sequences::update_visit_sequence (const Visit_seq & sequence, const size_t i_plan) [private]

Updates the visit sequence generate, mixing with pre-generated visit sequences.

Parameters

sequence

i_plan Update the visit sequence generate, mixing with pre-generated visit sequences.

Definition at line 317 of file Builder_visit_sequences.cpp.

References all_visit_seqs.

7.8.4 Member Data Documentation

7.8.4.1 genevalmag::Builder_visit_sequences::all_visit_seqs [private]

Stores all visit sequences generates.

Definition at line 49 of file Builder_visit_sequences.h.

Referenced by generate_visit_sequences(), get_visit_seq(), print_all_visit_sequences(), and update_visit_sequence().

7.8.4.2 genevalmag::Builder_visit_sequences::attr_grammar [private]

References to the attribute grammar.

Definition at line 43 of file Builder_visit_sequences.h.

Referenced by generate_visit_sequences().

7.8.4.3 genevalmag::Builder_visit_sequences::b_plans [private]

References to all plans generates.

Definition at line 37 of file Builder_visit_sequences.h.

Referenced by generate_visit_sequences().

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

- include/Builders/Builder_visit_sequences.h
- src/Builders/Builder_visit_sequences.cpp

7.9 genevalmag::c_rule Struct Reference

```
#include <Builder_plans.h>
```

Public Member Functions

- bool `operator< (const c_rule &other) const`
- `c_rule & operator= (const c_rule &other)`

Public Attributes

- unsigned short `father`
Index context.father (rule).
- `Order_rule context`
Indexes contexts inferior (rules).

7.9.1 Detailed Description

This struct represent a Context_rule, that is a father rule and the context of rule.

Definition at line 37 of file Builder_plans.h.

7.9.2 Member Function Documentation

7.9.2.1 bool genevalmag::c_rule::operator< (**const c_rule & other**) const [inline]

`operator<`

Parameters

other

Returns

Definition at line 54 of file Builder_plans.h.

References context, and father.

7.9.2.2 c_rule& genevalmag::c_rule::operator= (**const c_rule & other**) [inline]

`operator=`

Parameters

other

Returns

Definition at line 64 of file Builder_plans.h.

References context, and father.

7.9.3 Member Data Documentation

7.9.3.1 genevalmag::c_rule::context

Indexes contexts inferior (rules).

Definition at line 48 of file Builder_plans.h.

Referenced by genevalmag::Builder_plans::generate_plans(), genevalmag::Builder_plans::generates_topological_order(), operator<0, and operator=0.

7.9.3.2 genevalmag::c_rule::father

Index context father (rule).

Definition at line 43 of file Builder_plans.h.

Referenced by genevalmag::Builder_plans::generate_plans(), genevalmag::Builder_plans::generates_topological_order(), operator<0, and operator=0.

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

- include/Builders/Builder_plans.h

7.10 genevalmag::cycle_detector Struct Reference

Public Member Functions

- `cycle_detector (bool &has_cycle, Graph &graph)`
- `template<class Edge , class G >`
`void examine_edge (Edge u, const G &g)`
- `template<class Edge , class G >`
`void back_edge (Edge u, const G &g)`

Protected Attributes

- `bool & _has_cycle`
- `Graph & _graph_cycle`

7.10.1 Detailed Description

Definition at line 392 of file Builder_graphs.cpp.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 `genevalmag::cycle_detector::cycle_detector (bool & has_cycle, Graph & graph) [inline]`

Definition at line 395 of file Builder_graphs.cpp.

7.10.3 Member Function Documentation

7.10.3.1 `template<class Edge , class G > void genevalmag::cycle_detector::back_edge (Edge u, const G & g) [inline]`

Definition at line 430 of file Builder_graphs.cpp.

References `_has_cycle`.

7.10.3.2 `template<class Edge , class G > void genevalmag::cycle_detector::examine_edge (Edge u, const G & g) [inline]`

Re-define method for detect circle when examine each edge: Save ciclic subgraph.

Definition at line 401 of file Builder_graphs.cpp.

References `_graph_cycle`, `_has_cycle`, and `utilities::return_vertex()`.

Here is the call graph for this function:



7.10.4 Member Data Documentation

7.10.4.1 Graph& genevalmag::cycle_detector::_graph_cycle [protected]

Definition at line 437 of file Builder_graphs.cpp.

Referenced by examine_edge0.

7.10.4.2 bool& genevalmag::cycle_detector::_has_cycle [protected]

Definition at line 436 of file Builder_graphs.cpp.

Referenced by back_edge0, and examine_edge0.

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

- src/Builders/[Builder_graphs.cpp](#)

7.11 **genevalmag::decl_attribute Struct Reference**

Public Attributes

- `vector< string > d_names`
- `string d_sort_type`
- `type_attr d_mod_type`
- `string d_member_symbol`

7.11.1 Detailed Description

Type that represent the structure of a full declaration of one [Attribute](#).

Definition at line 54 of file Semantics_actions.cpp.

7.11.2 Member Data Documentation

7.11.2.1 **string genevalmag::decl_attribute::d_member_symbol**

Definition at line 59 of file Semantics_actions.cpp.

Referenced by `genevalmag::add_attribute()`, `genevalmag::create_attributes()`, and `genevalmag::save_member_list_attr()`.

7.11.2.2 **type_attr genevalmag::decl_attribute::d_mod_type**

Definition at line 58 of file Semantics_actions.cpp.

Referenced by `genevalmag::add_attribute()`, `genevalmag::create_attributes()`, and `genevalmag::save_type_attr()`.

7.11.2.3 **vector<string> genevalmag::decl_attribute::d_names**

Definition at line 56 of file Semantics_actions.cpp.

Referenced by `genevalmag::add_attribute()`, and `genevalmag::create_attributes()`.

7.11.2.4 **string genevalmag::decl_attribute::d_sort_type**

Definition at line 57 of file Semantics_actions.cpp.

Referenced by `genevalmag::create_attributes()`, and `genevalmag::save_sort_attr()`.

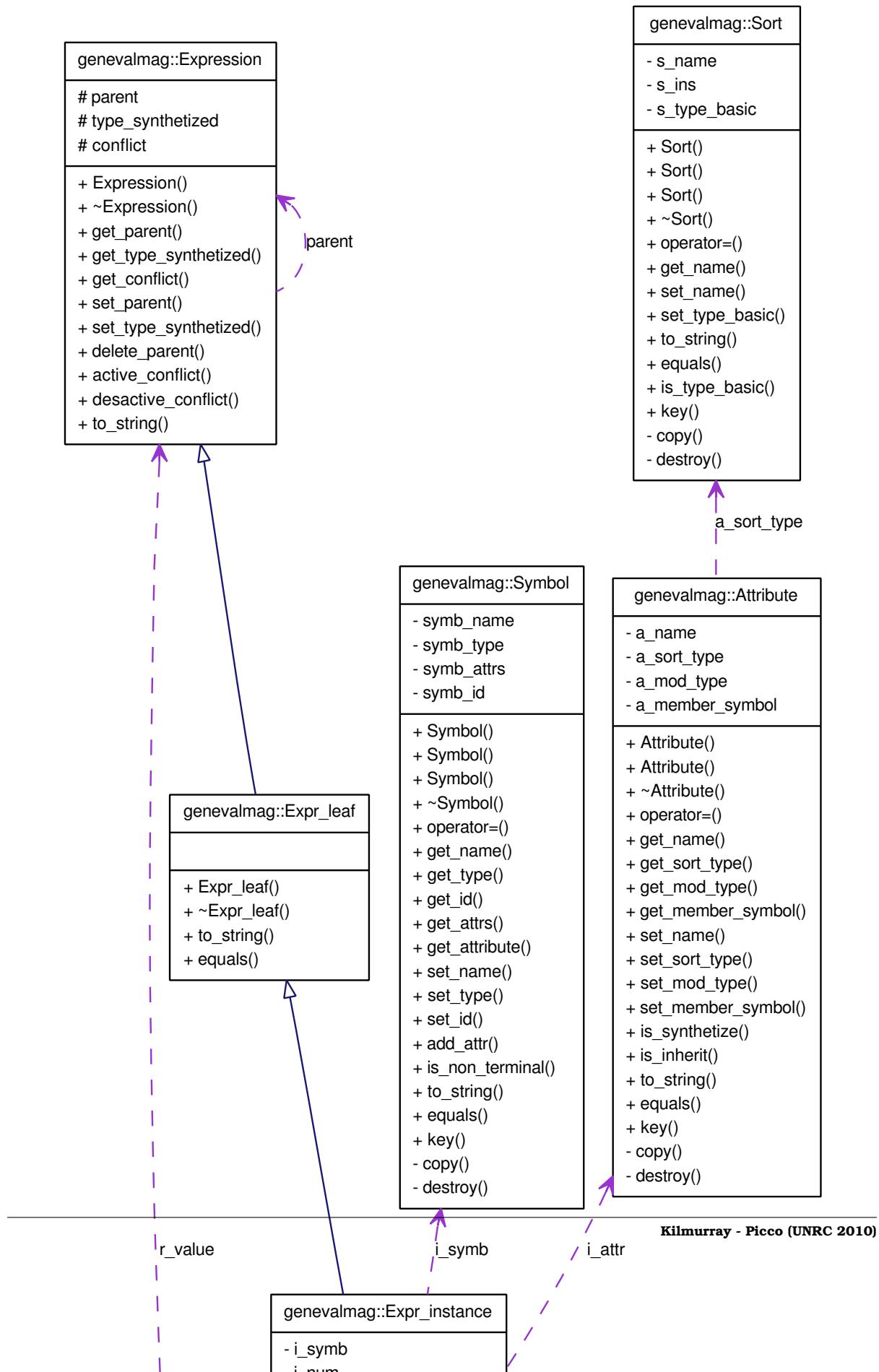
The documentation for this struct was generated from the following file:

- `src/Parser/Semantics_actions.cpp`

7.12 genevalmag::Equation Class Reference

```
#include <Equation.h>
```

Collaboration diagram for genevalmag::Equation:



Public Member Functions

- `Equation ()`
- `Equation (const Equation &other)`
- `virtual ~Equation ()`
- `Equation & operator= (const Equation &other)`
- `const Expr_instance * get_l_value () const`
- `const Expression * get_r_value () const`
- `const unsigned short get_id () const`
- `void set_id (const unsigned short id)`
- `void set_l_value (const Expr_instance &lvalue)`
- `void set_r_value (const Expression *rvalue)`
- `void inorder_only_leaf (const Expression *head, vector< const Expr_leaf * > &result) const`
- `const vector< const Expr_instance * > get_instance_right_side () const`
- `const string to_string () const`
- `const bool equals (const Equation &other) const`
- `const string key () const`

Private Member Functions

- `void copy (const Equation &other)`
- `void destroy ()`
- `unsigned short * _get_count_ref () const`
- `void inorder_only_instance (const Expression *head, vector< const Expr_instance * > &result) const`

Private Attributes

- `unsigned short eq_id`
Equation's identifier.
- `Expr_instance l_value`
- `const Expression * r_value`
- `unsigned short * count_ref`
Similar to a Smart Pointer to manage the release of the memory tree. When this is 0, all memory is freed from the tree.

7.12.1 Detailed Description

Definition at line 19 of file Equation.h.

7.12.2 Constructor & Destructor Documentation

7.12.2.1 genevalmag::Equation::Equation ()

Constructor empty of equation.

Definition at line 19 of file Equation.cpp.

7.12.2.2 genevalmag::Equation::Equation (const Equation & other)

Contructor copy of Equation.

Parameters

other

Returns

Contructor copy of Equation.

Definition at line 28 of file Equation.cpp.

7.12.2.3 genevalmag::Equation::~Equation () [virtual]

Destructor of equation.

Definition at line 36 of file Equation.cpp.

7.12.3 Member Function Documentation**7.12.3.1 unsigned short * genevalmag::Equation::_get_count_ref () const [private]**

Returns the count reference.

Returns

Return the count reference.

Definition at line 89 of file Equation.cpp.

7.12.3.2 void genevalmag::Equation::copy (const Equation & other) [private]

Method of copy the equation, STL-like C++.

Parameters

other Method of copy the equation, STL-like C++.

Definition at line 60 of file Equation.cpp.

7.12.3.3 void genevalmag::Equation::destroy () [private]

Method destroy equation, STL-like C++.

Definition at line 74 of file Equation.cpp.

7.12.3.4 const bool genevalmag::Equation::equals (const Equation & other) const

Compares the equation with other.

Parameters

other

Returns

Compares the equation with other.

Definition at line 245 of file Equation.cpp.

7.12.3.5 const unsigned short genevalmag::Equation::get_id () const

Returns the id of the equation.

Returns

Returns the id of the equation.

Definition at line 113 of file Equation.cpp.

7.12.3.6 const vector< const Expr_instance * > genevalmag::Equation::get_instance_right_side () const

Returns the pointer of the [Expression](#).

Returns

Returns the pointer of the [Expression](#).

Definition at line 213 of file Equation.cpp.

7.12.3.7 const Expr_instance * genevalmag::Equation::get_l_value () const

Returns the l_value of the equation.

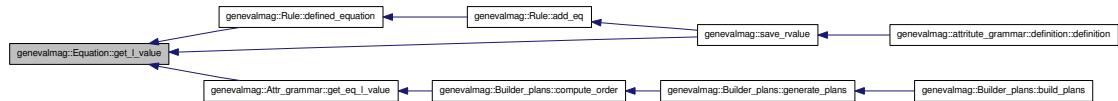
Returns

Returns the l_value of the equation.

Definition at line 97 of file Equation.cpp.

Referenced by genevalmag::Rule::defined_equation(), genevalmag::Attr_grammar::get_eq_l_value(), and genevalmag::save_rvalue().

Here is the caller graph for this function:



7.12.3.8 `const Expression * genevalmag::Equation::get_r_value () const`

Returns the r_value of the equation.

Parameters

Returns the r_value of the equation.

Definition at line 105 of file Equation.cpp.

7.12.3.9 `void genevalmag::Equation::inorder_only_instance (const Expression * head, vector< const Expr_instance * > & result) const [private]`

Traverse the equation tree while saves only the `Expr_instance` nodes in the vector result.

Parameters

`head`

`result` Traverse the equation tree while saves only the `Expr_instance` nodes in the vector result.

Definition at line 176 of file Equation.cpp.

7.12.3.10 `void genevalmag::Equation::inorder_only_leaf (const Expression * head, vector< const Expr_leaf * > & result) const`

Traverse the equation tree while saves only the `Expr_leaf` nodes in the vector result.

Parameters

`head`

`result` Traverse the equation tree while saves only the `Expr_leaf` nodes in the vector result.

Definition at line 145 of file Equation.cpp.

7.12.3.11 `const string genevalmag::Equation::key () const`

Generates and returns the string key that identifies an `Equation` definitely.

Result = <l_value><r_value>

where <l_value> = "instance_attr" and <r_value> is= "list of node_ast"

Returns

Generate and return the string key that identifies an [Equation](#) definitely.

Result = <l_value><r_value>

where <l_value> = "instance_attr" and <r_value> is= "list of node_ast"

Definition at line 257 of file Equation.cpp.

7.12.3.12 Equation & genevalmag::Equation::operator= (const Equation & other)

Operator assign(=) of [Equation](#).

Parameters

other

Returns

Operator assign(=) of [Equation](#).

Definition at line 47 of file Equation.cpp.

7.12.3.13 void genevalmag::Equation::set_id (const unsigned short id)

Sets the id of the equation.

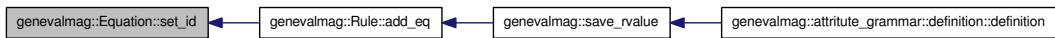
Parameters

id Sets the id of the equation.

Definition at line 121 of file Equation.cpp.

Referenced by [genevalmag::Rule::add_eq\(\)](#).

Here is the caller graph for this function:

**7.12.3.14 void genevalmag::Equation::set_l_value (const Expr_instance & lvalue)**

Sets the left value of the equation.

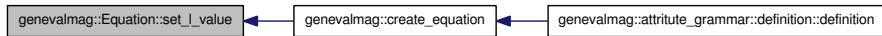
Parameters

lvalue Set the left value of the equation.

Definition at line 129 of file Equation.cpp.

Referenced by [genevalmag::create_equation\(\)](#).

Here is the caller graph for this function:



7.12.3.15 void genevalmag::Equation::set_r_value (const Expression * rvalue)

Sets the right value of the equation: is a tree.

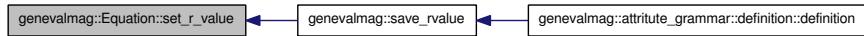
Parameters

rvalue Set the left value of the equation.

Definition at line 137 of file Equation.cpp.

Referenced by genevalmag::save_rvalue().

Here is the caller graph for this function:



7.12.3.16 const string genevalmag::Equation::to_string () const

Generates and returns a string representation of an Equation.

Result = <l_value> "=" <r_value> ";"

where <l_value> = "instance_attr" and <r_value> is= "list of node_ast"

Returns

Generate and return a string representation of an Equation.

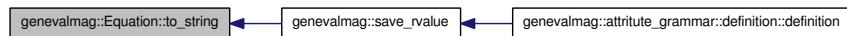
Result= <l_value> "=" <r_value> ";"

where <l_value> = "instance_attr" and <r_value> is= "list of node_ast"

Definition at line 227 of file Equation.cpp.

Referenced by genevalmag::save_rvalue().

Here is the caller graph for this function:



7.12.4 Member Data Documentation

7.12.4.1 genevalmag::Equation::count_ref [private]

Similar to a Smart Pointer to manage the release of the memory tree. When this is 0, all memory is freed from the tree.

Definition at line 43 of file Equation.h.

7.12.4.2 **genevalmag::Equation::eq_id [private]**

Equation's identifier.

Equation's tree expression.

Equation's left value.

Definition at line 26 of file Equation.h.

7.12.4.3 **Expr_instance genevalmag::Equation::l_value [private]**

Definition at line 31 of file Equation.h.

7.12.4.4 **const Expression* genevalmag::Equation::r_value [private]**

Definition at line 36 of file Equation.h.

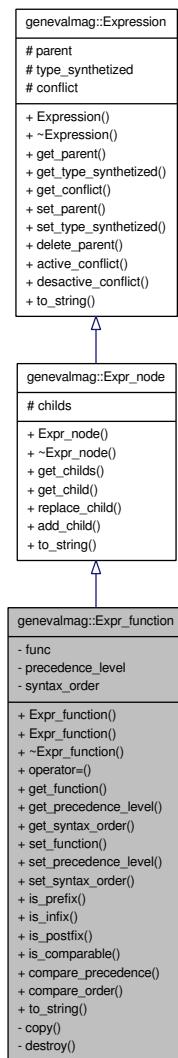
The documentation for this class was generated from the following files:

- include/Attr_grammar/Equation.h
- src/Attr_grammar/Equation.cpp

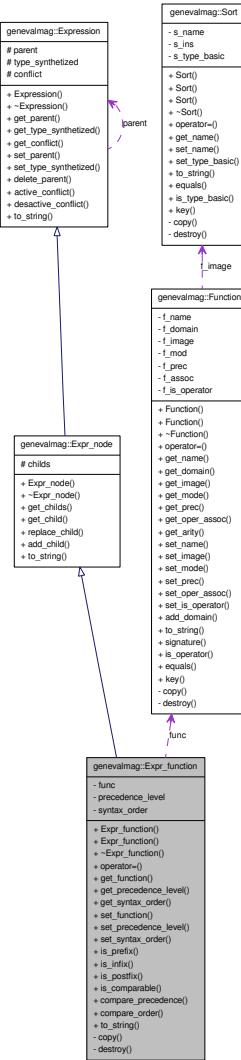
7.13 genevalmag::Expr_function Class Reference

```
#include <Expr_function.h>
```

Inheritance diagram for genevalmag::Expr_function:



Collaboration diagram for genevalmag::Expr_function:



Public Member Functions

- `Expr_function()`
- `Expr_function(const Expr_function &other)`
- `virtual ~Expr_function()`
- `Expr_function & operator=(const Expr_function &other)`
- `const Function * get_function() const`
- `unsigned short get_precedence_level() const`
- `unsigned short get_syntax_order() const`
- `void set_function(const Function *function)`
- `void set_precedence_level(unsigned short p_level)`
- `void set_syntax_order(unsigned short s_order)`
- `bool is_prefix() const`
- `bool is_infix() const`

- bool `is_postfix () const`
- bool `is_comparable (const Expr_function *other) const`
- int `compare_precedence (const Expr_function *other) const`
- int `compare_order (const Expr_function *other) const`
- string `to_string () const`

Private Member Functions

- void `copy (const Expr_function &other)`
- void `destroy ()`

Private Attributes

- const Function * `func`
Function associates with Expr_function.
- unsigned short `precedence_level`
Level precedence of the function.
- unsigned short `syntax_order`
Syntax order in the parse.

7.13.1 Detailed Description

Definition at line 23 of file Expr_function.h.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 genevalmag::Expr_function::Expr_function ()

Constructor empty of Expr_function.

Returns

Constructor empty of Expr_function.

Definition at line 24 of file Expr_function.cpp.

References precedence_level, and syntax_order.

7.13.2.2 genevalmag::Expr_function::Expr_function (const Expr_function & other)

Constructor copy of Expr_function.

Parameters

other

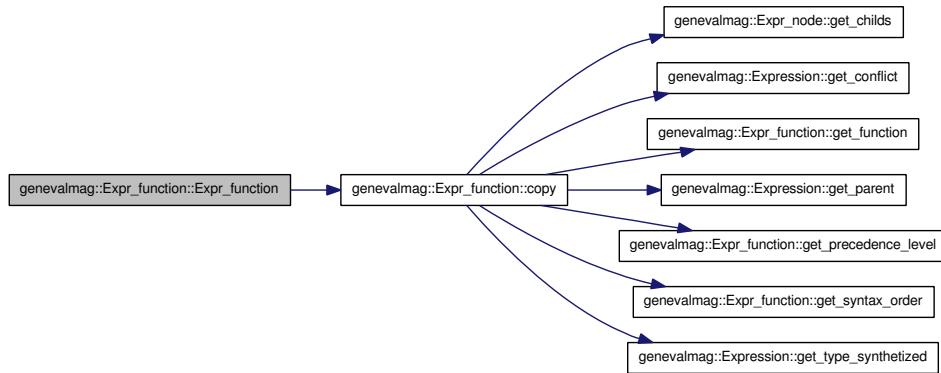
Returns

Constructor copy of [Expr_function](#).

Definition at line 37 of file Expr_function.cpp.

References [copy\(\)](#).

Here is the call graph for this function:

**7.13.2.3 genevalmag::Expr_function::~Expr_function () [virtual]**

Destructor of [Expr_function](#).

Returns

Destructor of [Expr_function](#).

Definition at line 49 of file Expr_function.cpp.

References [destroy\(\)](#).

Here is the call graph for this function:

**7.13.3 Member Function Documentation****7.13.3.1 int genevalmag::Expr_function::compare_order (const Expr_function * other) const**

Returns:

≥ 0 when is same order

< 0 when other have great order

> 0 when other have small order

Parameters*other***Returns**

Return: = 0 when is same order < 0 when other have great order > 0 when other have small order

Definition at line 193 of file Expr_function.cpp.

References get_syntax_order(), and syntax_order.

Here is the call graph for this function:



7.13.3.2 int genevalmag::Expr_function::compare_precedence (const Expr_function * other) const

Returns:

\= 0 when is same precedence

< 0 when other have great precedence

> 0 when other have small precedence

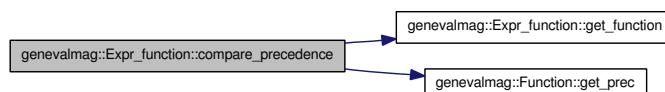
Parameters*other***Returns**

Return: = 0 when is same precedence < 0 when other have great precedence > 0 when other have small precedence

Definition at line 182 of file Expr_function.cpp.

References func, get_function(), and genevalmag::Function::get_prec().

Here is the call graph for this function:



7.13.3.3 void genevalmag::Expr_function::copy (const Expr_function & other) [private]

Method of copy the Expr_function, STL-like C++.

Parameters

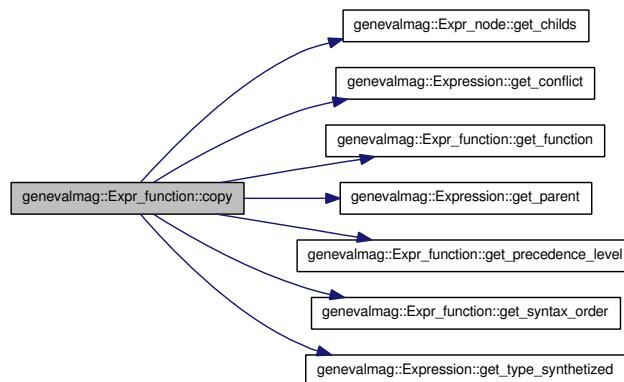
other Method of copy the Expr_function, STL-like C++.

Definition at line 78 of file Expr_function.cpp.

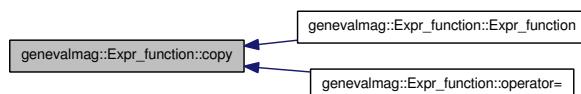
References genevalmag::Expr_node::childs, genevalmag::Expression::conflict, func, genevalmag::Expr_node::get_childs(), genevalmag::Expression::get_conflict(), get_function(), genevalmag::Expression::get_parent(), get_precedence_level(), get_syntax_order(), genevalmag::Expression::get_type_synthesized(), genevalmag::Expression::parent, precedence_level, syntax_order, and genevalmag::Expression::type_synthesized.

Referenced by Expr_function(), and operator=().

Here is the call graph for this function:



Here is the caller graph for this function:



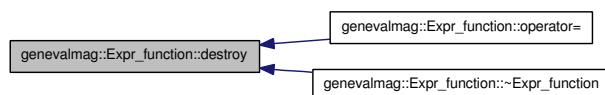
7.13.3.4 void genevalmag::Expr_function::destroy () [private]

Method destroy Expr_function, STL-like C++.

Definition at line 92 of file Expr_function.cpp.

Referenced by operator=(), and ~Expr_function().

Here is the caller graph for this function:



7.13.3.5 const Function * genevalmag::Expr_function::get_function () const

Returns the function pointer of the [Expr_function](#).

Returns

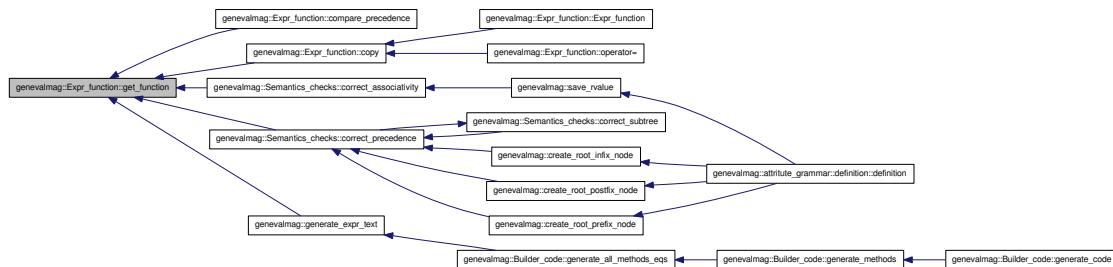
Returns the function pointer of the [Expr_function](#).

Definition at line 99 of file Expr_function.cpp.

References func.

Referenced by compare_precedence(), copy(), genevalmag::Semantics_checks::correct_associativity(), genevalmag::Semantics_checks::correct_precedence(), and genevalmag::generate_expr_text().

Here is the caller graph for this function:



7.13.3.6 unsigned short genevalmag::Expr_function::get_precedence_level () const

Returns the precedence level of the [Expr_function](#).

Returns

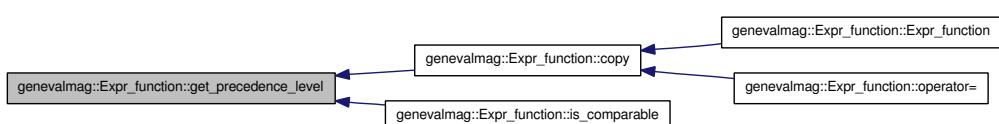
Returns the precedence level of the [Expr_function](#).

Definition at line 107 of file Expr_function.cpp.

References precedence_level.

Referenced by copy(), and is_comparable().

Here is the caller graph for this function:



7.13.3.7 unsigned short genevalmag::Expr_function::get_syntax_order () const

Returns the syntax order of the [Expr_function](#).

Returns

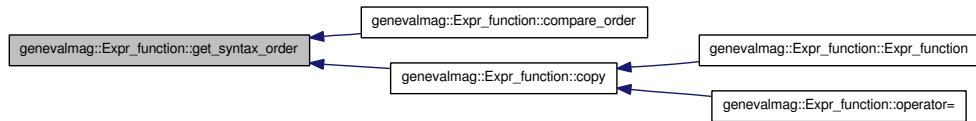
Returns the syntax order of the [Expr_function](#).

Definition at line 115 of file Expr_function.cpp.

References [syntax_order](#).

Referenced by [compare_order\(\)](#), and [copy\(\)](#).

Here is the caller graph for this function:

**7.13.3.8 bool genevalmag::Expr_function::is_comparable (const Expr_function * other) const**

Returns if both Expr_functions are in the same precedence level.

Parameters

other

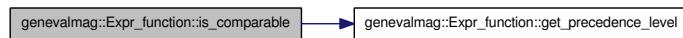
Returns

Returns if both Expr_functions are in the same precedence level.

Definition at line 171 of file Expr_function.cpp.

References [get_precedence_level\(\)](#), and [precedence_level](#).

Here is the call graph for this function:

**7.13.3.9 bool genevalmag::Expr_function::is_infix () const**

Returns if the function of [Expr_function](#) is infix.

Returns

Returns if the function of [Expr_function](#) is infix.

Definition at line 155 of file [Expr_function.cpp](#).

References [func](#), [genevalmag::Function::get_mode\(\)](#), and [genevalmag::k_infix](#).

Referenced by [genevalmag::generate_expr_text\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.13.3.10 bool genevalmag::Expr_function::is_postfix () const

Returns if the function of [Expr_function](#) is postfix.

Returns

Returns if the function of [Expr_function](#) is postfix.

Definition at line 163 of file [Expr_function.cpp](#).

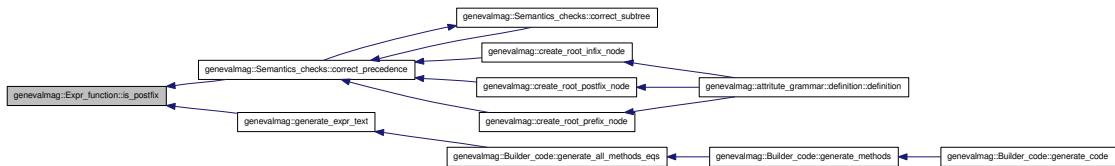
References [func](#), [genevalmag::Function::get_mode\(\)](#), and [genevalmag::k_postfix](#).

Referenced by [genevalmag::Semantics_checks::correct_precedence\(\)](#), and [genevalmag::generate_expr_text\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.13.3.11 bool genevalmag::Expr_function::is_prefix () const

Returns if the function of [Expr_function](#) is prefix.

Returns

Returns if the function of [Expr_function](#) is prefix.

Definition at line 147 of file Expr_function.cpp.

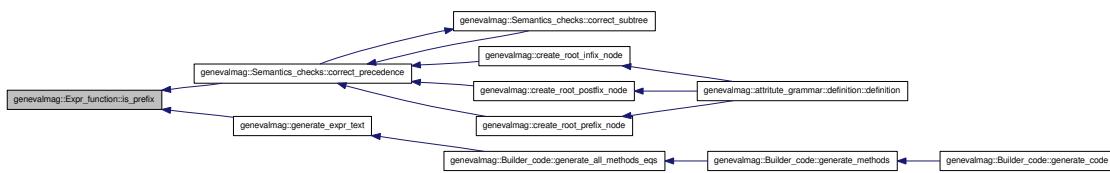
References func, genevalmag::Function::get_mode(), and genevalmag::k_prefix.

Referenced by genevalmag::Semantics_checks::correct_precedence(), and genevalmag::generate_expr_text().

Here is the call graph for this function:



Here is the caller graph for this function:



7.13.3.12 Expr_function & genevalmag::Expr_function::operator= (const Expr_function & other)

Operator assign(=) of [Expr_function](#).

Parameters

other

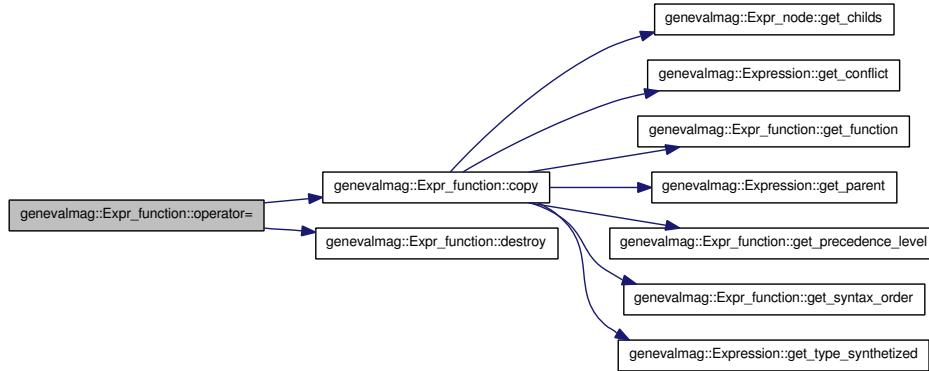
Returns

Operator assign(=) of [Expr_function](#).

Definition at line 65 of file Expr_function.cpp.

References copy(), and destroy().

Here is the call graph for this function:



7.13.3.13 void genevalmag::Expr_function::set_function (const Function *function)

Sets the function pointer of the [Expr_function](#).

Parameters

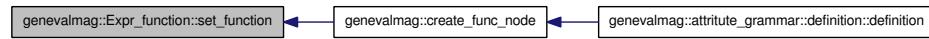
function Sets the function pointer of the [Expr_function](#).

Definition at line 123 of file Expr_function.cpp.

References func.

Referenced by genevalmag::create_func_node0.

Here is the caller graph for this function:



7.13.3.14 void genevalmag::Expr_function::set_precedence_level (unsigned short p_level)

Sets the precedence level of the [Expr_function](#).

Parameters

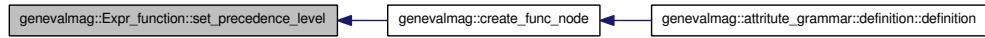
p_level Sets the precedence level of the [Expr_function](#).

Definition at line 131 of file Expr_function.cpp.

References precedence_level.

Referenced by genevalmag::create_func_node0.

Here is the caller graph for this function:



7.13.3.15 void genevalmag::Expr_function::set_syntax_order (unsigned short s_order)

Sets the syntax order of the [Expr_function](#).

Parameters

s_order Sets the syntax order of the [Expr_function](#).

Definition at line 139 of file Expr_function.cpp.

References [syntax_order](#).

Referenced by [genevalmag::create_func_node\(\)](#).

Here is the caller graph for this function:

**7.13.3.16 string genevalmag::Expr_function::to_string () const [virtual]**

Generate and return a string representation of a [Expr_function](#).

If is a pure function:

Result= <func_name> "(" child[0], ... , child[n] ")"

If is an infix binary operator:

Result= "(" child[0] <op_name> child[1] ")"

If is an prefix unary operator:

Result= <op_name> "(" child[0] ")"

If is an postfix unary operator:

Result= "(" child[0] ")" <op_name>

Returns

Generate and return a string representation of a [Expr_function](#).

If is a pure function: Result= <func_name> "(" child[0], ... , child[n] ")"

If is an infix binary operator: Result= "(" child[0] <op_name> child[1] ")"

If is an prefix unary operator: Result= <op_name> "(" child[0] ")"

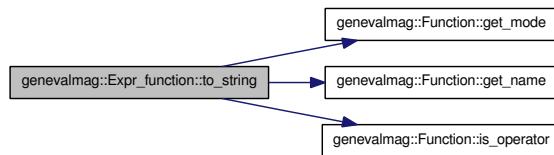
If is an postfix unary operator: Result= "(" child[0] ")" <op_name>

Implements [genevalmag::Expr_node](#).

Definition at line 213 of file Expr_function.cpp.

References [genevalmag::Expr_node::childs](#), [func](#), [genevalmag::Function::get_mode\(\)](#), [genevalmag::Function::get_name\(\)](#), [genevalmag::Function::is_operator\(\)](#), [genevalmag::k_infix](#), [genevalmag::k_postfix](#), and [genevalmag::k_prefix](#).

Here is the call graph for this function:



7.13.4 Member Data Documentation

7.13.4.1 `genevalmag::Expr_function::func` [private]

Function associates with [Expr_function](#).

Definition at line 30 of file Expr_function.h.

Referenced by compare_precedence(), copy(), get_function(), is_infix(), is_postfix(), is_prefix(), set_function(), and to_string().

7.13.4.2 `genevalmag::Expr_function::precedence_level` [private]

Level precedence of the function.

Definition at line 35 of file Expr_function.h.

Referenced by copy(), Expr_function(), get_precedence_level(), is_comparable(), and set_precedence_level().

7.13.4.3 `genevalmag::Expr_function::syntax_order` [private]

Syntax order in the parse.

Definition at line 40 of file Expr_function.h.

Referenced by compare_order(), copy(), Expr_function(), get_syntax_order(), and set_syntax_order().

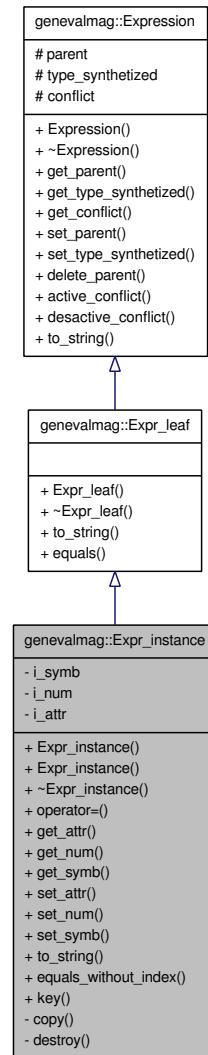
The documentation for this class was generated from the following files:

- include/Expression_tree/[Expr_function.h](#)
- src/Expression_tree/[Expr_function.cpp](#)

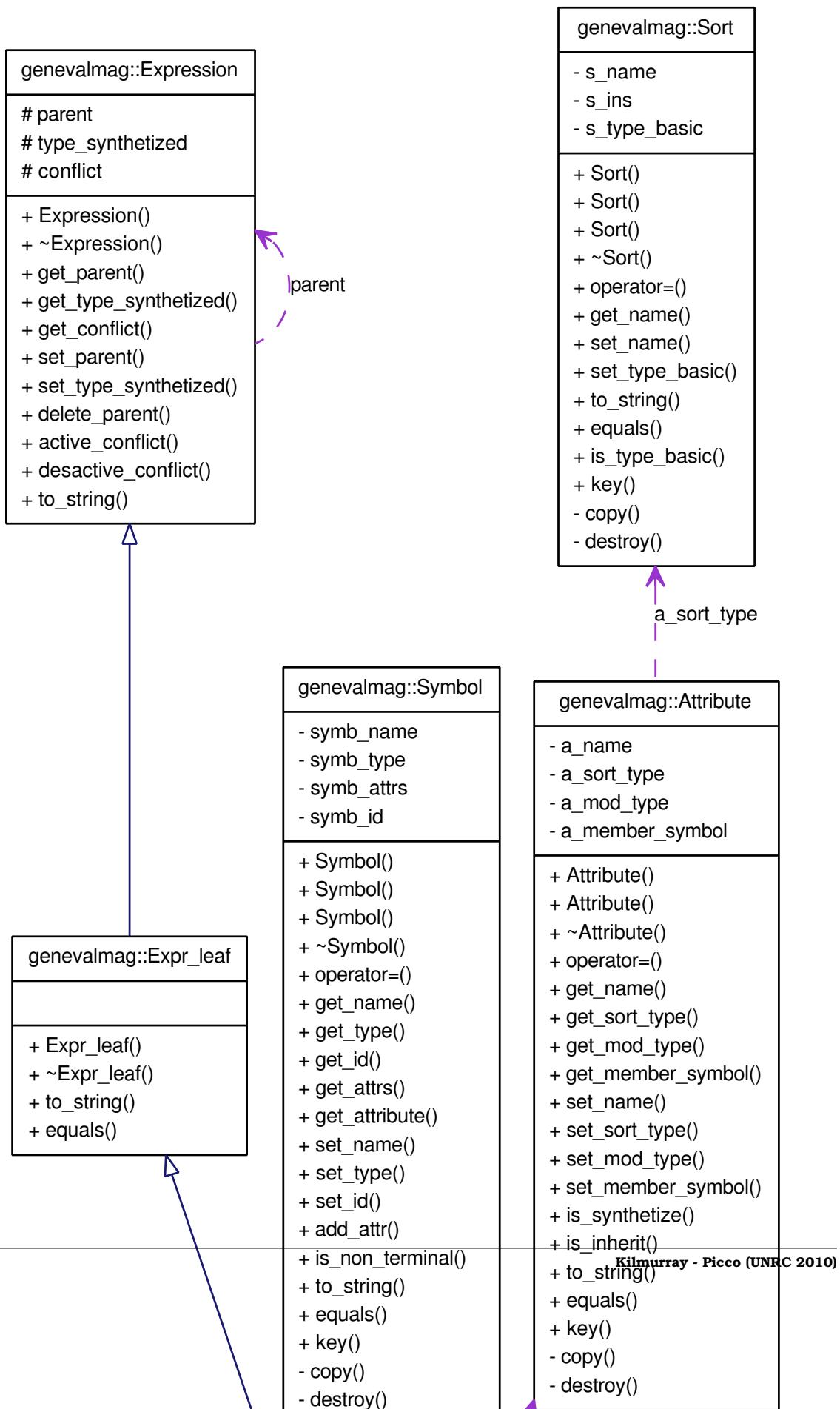
7.14 genevalmag::Expr_instance Class Reference

```
#include <Expr_instance.h>
```

Inheritance diagram for genevalmag::Expr_instance:



Collaboration diagram for genevalmag::Expr_instance:



Public Member Functions

- `Expr_instance()`
- `Expr_instance(const Expr_instance &other)`
- `virtual ~Expr_instance()`
- `Expr_instance & operator=(const Expr_instance &other)`
- `const Attribute * get_attr() const`
- `unsigned short get_num() const`
- `const Symbol * get_symb() const`
- `void set_attr(const Attribute *attr)`
- `void set_num(unsigned short num)`
- `void set_symb(const Symbol *symb)`
- `string to_string() const`
- `bool equals_without_index(const Expr_instance *other) const`
- `string key() const`

Private Member Functions

- `void copy(const Expr_instance &other)`
- `void destroy()`

Private Attributes

- `const Symbol * i_symb`
Symbol of Expr_instance.
- `unsigned short i_num`
Syntax index of the instance.
- `const Attribute * i_attr`
Attribute of Expr_instance.

7.14.1 Detailed Description

Definition at line 20 of file Expr_instance.h.

7.14.2 Constructor & Destructor Documentation

7.14.2.1 genevalmag::Expr_instance::Expr_instance()

Constructor empty of `Expr_instance`.

Returns

Constructor empty of `Expr_instance`.

Definition at line 25 of file Expr_instance.cpp.

7.14.2.2 `genevalmag::Expr_instance::Expr_instance (const Expr_instance & other)`

Constructor copy of `Expr_instance`.

Parameters

other

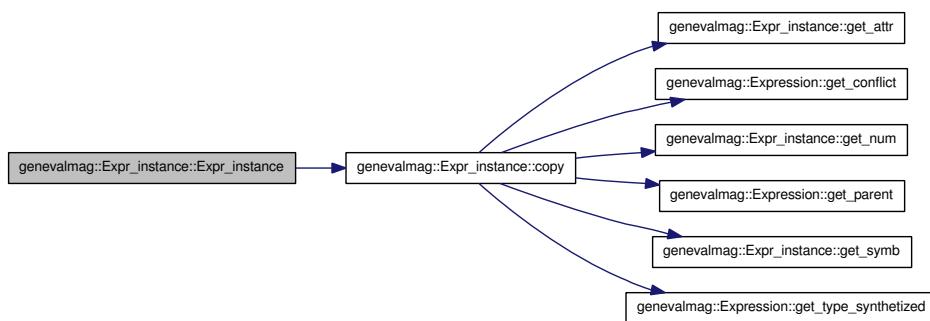
Returns

Constructor copy of `Expr_instance`.

Definition at line 35 of file `Expr_instance.cpp`.

References `copy()`.

Here is the call graph for this function:



7.14.2.3 `genevalmag::Expr_instance::~Expr_instance () [virtual]`

Destructor of `Expr_instance`.

Returns

Destructor of `Expr_instance`.

Definition at line 47 of file `Expr_instance.cpp`.

References `destroy()`.

Here is the call graph for this function:



7.14.3 Member Function Documentation

7.14.3.1 void genevalmag::Expr_instance::copy (const Expr_instance & other) [private]

Method of copy the [Expr_instance](#), STL-like C++.

Parameters

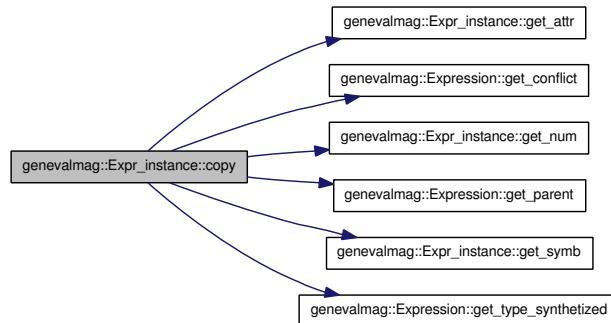
other Method of copy the [Expr_instance](#), STL-like C++.

Definition at line 76 of file [Expr_instance.cpp](#).

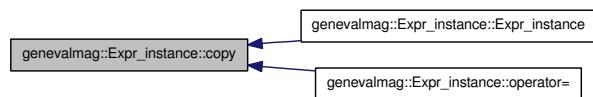
References [genevalmag::Expression::conflict](#), [get_attr\(\)](#), [genevalmag::Expression::get_conflict\(\)](#), [get_num\(\)](#), [genevalmag::Expression::get_parent\(\)](#), [get_symb\(\)](#), [genevalmag::Expression::get_type_synthesized\(\)](#), [i_attr](#), [i_num](#), [i_symb](#), [genevalmag::Expression::parent](#), and [genevalmag::Expression::type_synthesized](#).

Referenced by [Expr_instance\(\)](#), and [operator=\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.14.3.2 void genevalmag::Expr_instance::destroy () [private]

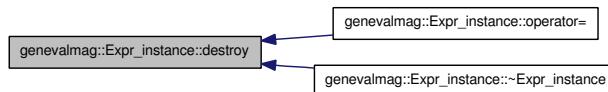
Method destroy [Expr_instance](#), STL-like C++.

Definition at line 89 of file [Expr_instance.cpp](#).

References [i_attr](#), and [i_symb](#).

Referenced by [operator=\(\)](#), and [~Expr_instance\(\)](#).

Here is the caller graph for this function:



7.14.3.3 bool genevalmag::Expr_instance::equals_without_index (const Expr_instance * other) const

Compares the `Expr_instance` with other.

Respects `Symbol` and attribute.

Parameters

other

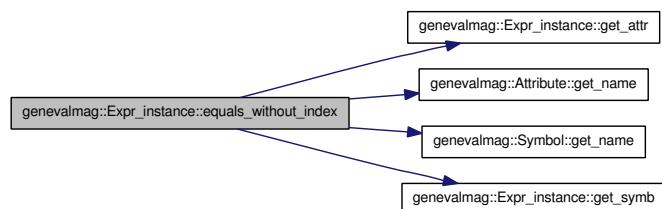
Returns

Compares the `Expr_instance` with other. Respects `Symbol` and attribute.

Definition at line 172 of file `Expr_instance.cpp`.

References `get_attr()`, `genevalmag::Attribute::get_name()`, `genevalmag::Symbol::get_name()`, `get_symb()`, `i_attr`, and `i_symb`.

Here is the call graph for this function:



7.14.3.4 const Attribute * genevalmag::Expr_instance::get_attr () const

Returns the attribute pointer of the `Expr_instance`.

Returns

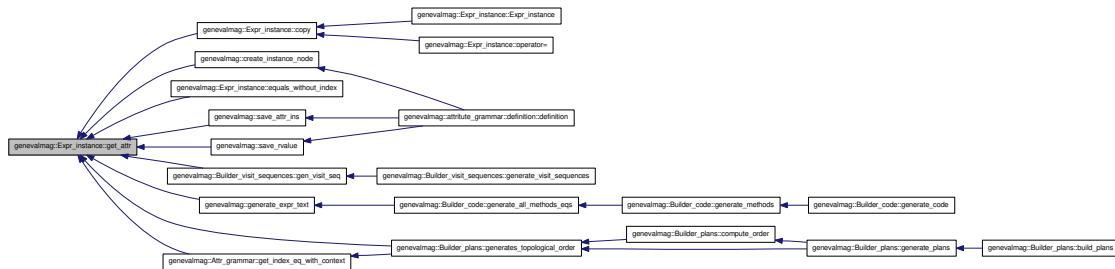
Returns the attribute pointer of the `Expr_instance`.

Definition at line 98 of file `Expr_instance.cpp`.

References `i_attr`.

Referenced by `copy()`, `genevalmag::create_instance_node()`, `equals_without_index()`, `genevalmag::Builder_visit_sequences::gen_visit_seq()`, `genevalmag::generate_expr_text()`, `genevalmag::Builder_plans::generates_topological_order()`, `genevalmag::Attr_grammar::get_index_eq_with_context()`, `genevalmag::save_attr_ins()`, and `genevalmag::save_rvalue()`.

Here is the caller graph for this function:



7.14.3.5 unsigned short genevalmag::Expr_instance::get_num () const

Returns the number of the `Expr_instance`.

Returns

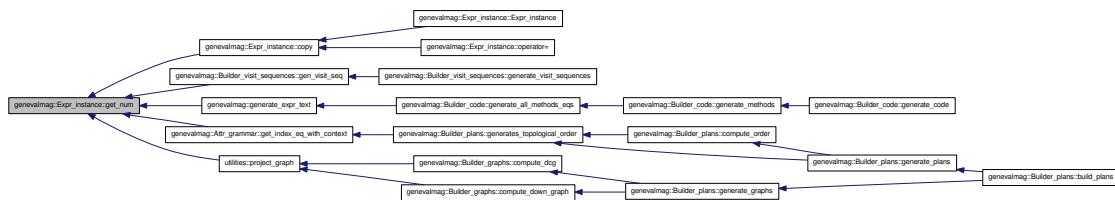
Returns the number of the `Expr_instance`.

Definition at line 106 of file `Expr_instance.cpp`.

References `i_num`.

Referenced by `copy()`, `genevalmag::Builder_visit_sequences::gen_visit_seq()`, `genevalmag::generate_expr_text()`, `genevalmag::Attr_grammar::get_index_eq_with_context()`, and `utilities::project_graph()`.

Here is the caller graph for this function:



7.14.3.6 const Symbol * genevalmag::Expr_instance::get_symb () const

Returns the symbol pointer of the `Expr_instance`.

Returns

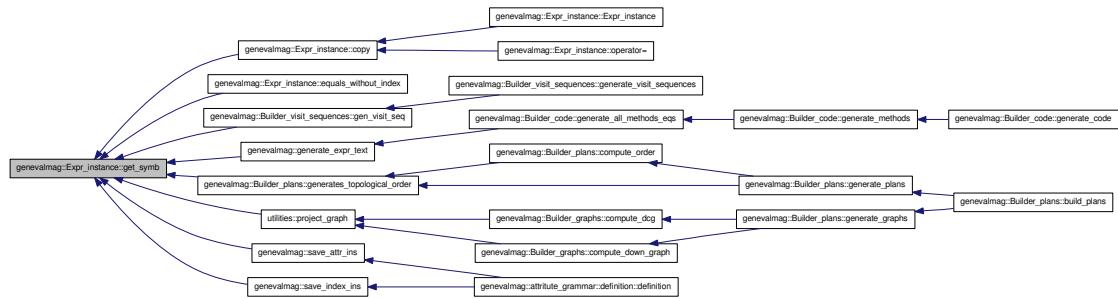
Returns the symbol pointer of the `Expr_instance`.

Definition at line 114 of file `Expr_instance.cpp`.

References `i_symb`.

Referenced by `copy()`, `equals_without_index()`, `genevalmag::Builder_visit_sequences::gen_visit_seq()`, `genevalmag::generate_expr_text()`, `genevalmag::Builder_plans::generates_topological_order()`, `utilities::project_graph()`, `genevalmag::save_attr_ins()`, and `genevalmag::save_index_ins()`.

Here is the caller graph for this function:



7.14.3.7 string `genevalmag::Expr_instance::key () const`

Generates and returns the string key that identifies an `Expr_instance` definitely.

Result= <symbol><number><attribute>

Ex: E0valor

Returns

Generate and return the string key that identifies an `Expr_instance` definitely.

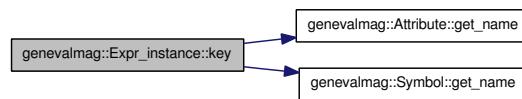
Result= <symbol><number><attribute>

Ex: E0valor

Definition at line 192 of file `Expr_instance.cpp`.

References `genevalmag::Attribute::get_name()`, `genevalmag::Symbol::get_name()`, `i_attr`, `i_num`, and `i_symb`.

Here is the call graph for this function:



7.14.3.8 Expr_instance & genevalmag::Expr_instance::operator= (const Expr_instance & other)

Operator assign(=) of Expr_instance.

Parameters

other

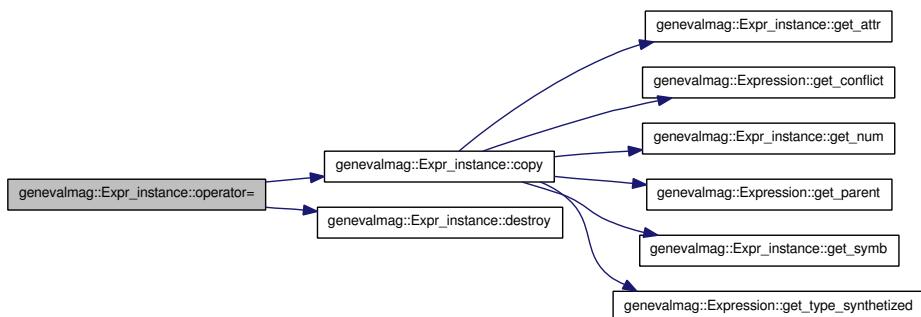
Returns

Operator assign(=) of Expr_instance.

Definition at line 63 of file Expr_instance.cpp.

References copy(), and destroy().

Here is the call graph for this function:



7.14.3.9 void genevalmag::Expr_instance::set_attr (const Attribute * attr)

Sets the attribute pointer of the Expr_instance.

Parameters

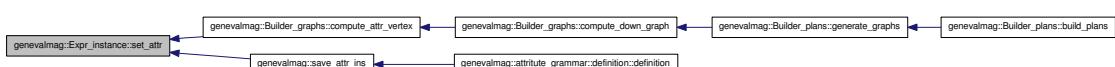
attr Sets the attribute pointer of the Expr_instance.

Definition at line 122 of file Expr_instance.cpp.

References i_attr.

Referenced by genevalmag::Builder_graphs::compute_attr_vertex(), and genevalmag::save_attr_ins().

Here is the caller graph for this function:



7.14.3.10 void genevalmag::Expr_instance::set_num (unsigned short num)

Sets the number of the [Expr_instance](#).

Parameters

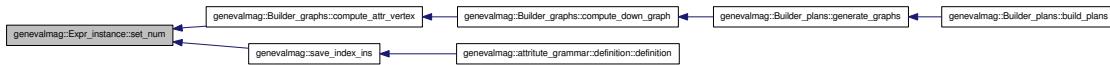
num Sets the number of the [Expr_instance](#).

Definition at line 130 of file Expr_instance.cpp.

References i_num.

Referenced by `genevalmag::Builder_graphs::compute_attr_vertex()`, and `genevalmag::save_index_ins()`.

Here is the caller graph for this function:



7.14.3.11 void genevalmag::Expr_instance::set_symb (const Symbol * symb)

Sets the symbol pointer of the [Expr_instance](#).

Parameters

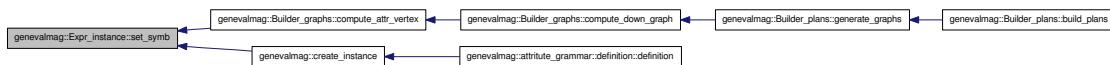
symb Sets the symbol pointer of the [Expr_instance](#).

Definition at line 138 of file Expr_instance.cpp.

References i_symb.

Referenced by `genevalmag::Builder_graphs::compute_attr_vertex()`, and `genevalmag::create_instance()`.

Here is the caller graph for this function:



7.14.3.12 string genevalmag::Expr_instance::to_string () const [virtual]

Generates and returns a string representation of a [Expr_instance](#).

Result= <symbol>"["<number>"]."<attribute>

Ex: E[0].valor

Returns

Generate and return a string representation of a [Expr_instance](#).

Result= <symbol>"["<number>"].<attribute>

Ex: E[0].valor

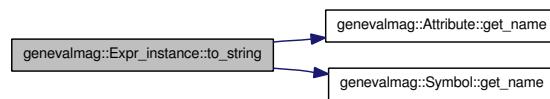
Implements [genevalmag::Expr_leaf](#).

Definition at line 150 of file Expr_instance.cpp.

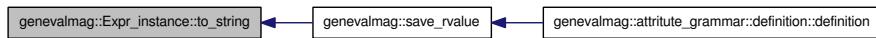
References [genevalmag::Attribute::get_name\(\)](#), [genevalmag::Symbol::get_name\(\)](#), [i_attr](#), [i_num](#), and [i_symb](#).

Referenced by [genevalmag::save_rvalue\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.14.4 Member Data Documentation

7.14.4.1 genevalmag::Expr_instance::i_attr [private]

Attribute of [Expr_instance](#).

Definition at line 37 of file Expr_instance.h.

Referenced by [copy\(\)](#), [destroy\(\)](#), [equals_without_index\(\)](#), [get_attr\(\)](#), [key\(\)](#), [set_attr\(\)](#), and [to_string\(\)](#).

7.14.4.2 genevalmag::Expr_instance::i_num [private]

Syntax index of the instance.

Definition at line 32 of file Expr_instance.h.

Referenced by [copy\(\)](#), [get_num\(\)](#), [key\(\)](#), [set_num\(\)](#), and [to_string\(\)](#).

7.14.4.3 genevalmag::Expr_instance::i_symb [private]

Symbol of [Expr_instance](#).

Definition at line 27 of file Expr_instance.h.

Referenced by [copy\(\)](#), [destroy\(\)](#), [equals_without_index\(\)](#), [get_symb\(\)](#), [key\(\)](#), [set_symb\(\)](#), and [to_string\(\)](#).

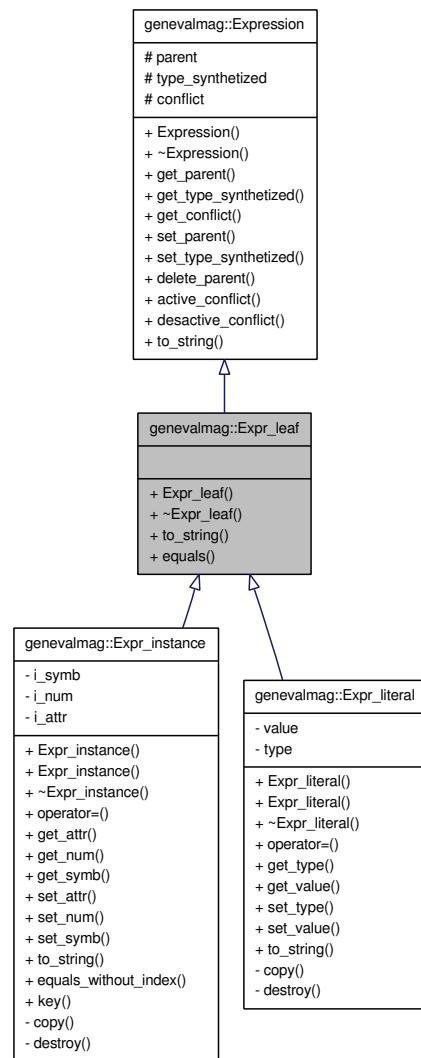
The documentation for this class was generated from the following files:

- include/Expression_tree/Expr_instance.h
- src/Expression_tree/Expr_instance.cpp

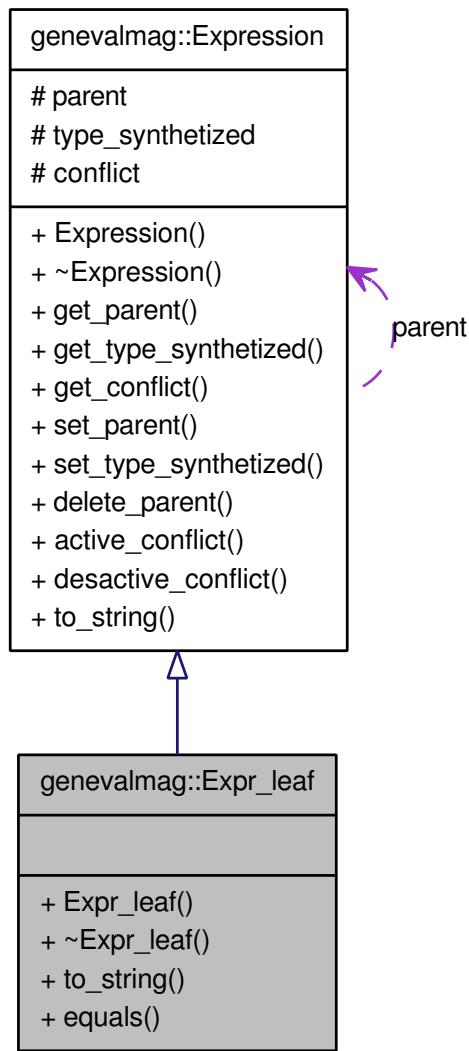
7.15 genevalmag::Expr_leaf Class Reference

```
#include <Expr_leaf.h>
```

Inheritance diagram for genevalmag::Expr_leaf:



Collaboration diagram for genevalmag::Expr_leaf:



Public Member Functions

- `Expr_leaf()`
- `virtual ~Expr_leaf()`
- `virtual string to_string() const =0`
- `bool equals(const Expr_leaf *other) const`

7.15.1 Detailed Description

Definition at line 17 of file Expr_leaf.h.

7.15.2 Constructor & Destructor Documentation

7.15.2.1 genevalmag::Expr_leaf::Expr_leaf()

Constructor of [Expr_leaf](#).

Returns

Definition at line 14 of file Expr_leaf.cpp.

7.15.2.2 genevalmag::Expr_leaf::~Expr_leaf() [virtual]

Destructor of [Expr_leaf](#).

Returns

Destructor of [Expr_leaf](#).

Definition at line 19 of file Expr_leaf.cpp.

7.15.3 Member Function Documentation

7.15.3.1 bool genevalmag::Expr_leaf::equals (const Expr_leaf * *other*) const

Compares the [Expr_leaf](#) with other.

Parameters

other

Returns

Compares the [Expr_leaf](#) with other.

Definition at line 24 of file Expr_leaf.cpp.

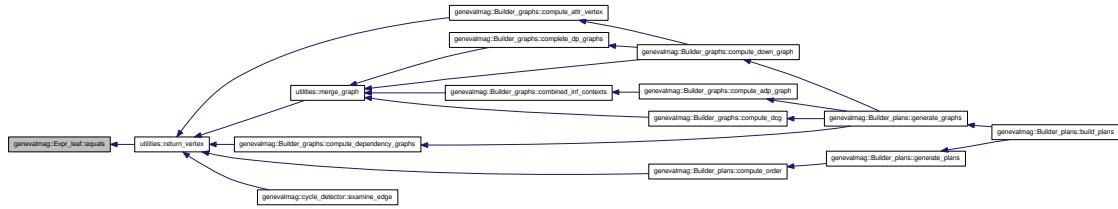
References to_string().

Referenced by utilities::return_vertex().

Here is the call graph for this function:



Here is the caller graph for this function:



7.15.3.2 `virtual string genevalmag::Expr_leaf::to_string() const [pure virtual]`

Generates and returns a string representation of a `Expr_leaf`.

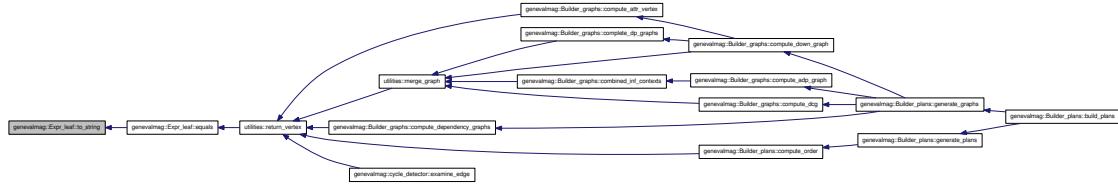
Returns

Implements `genevalmag::Expression`.

Implemented in `genevalmag::Expr_instance`, and `genevalmag::Expr_literal`.

Referenced by `equals()`.

Here is the caller graph for this function:



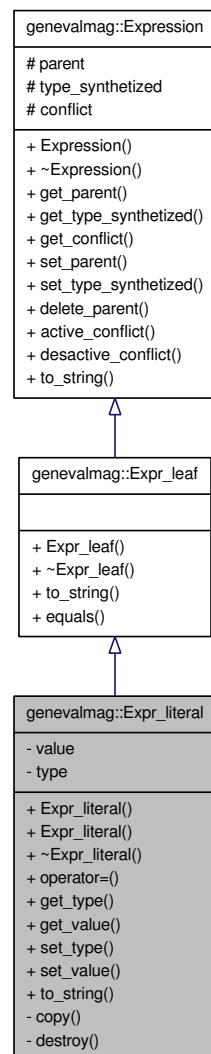
The documentation for this class was generated from the following files:

- include/Expression_tree/`Expr_leaf.h`
- src/Expression_tree/`Expr_leaf.cpp`

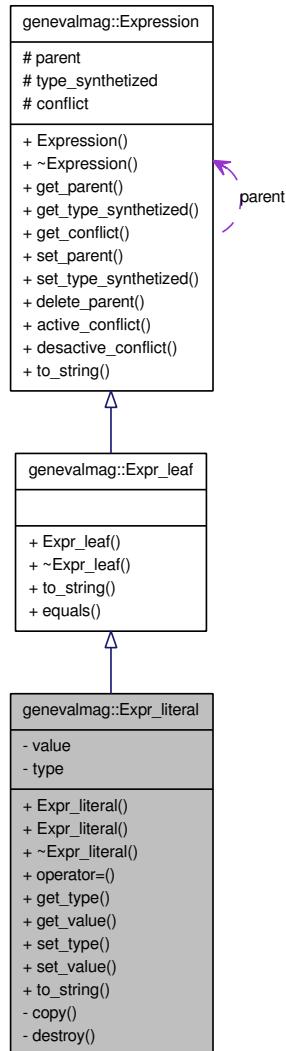
7.16 genevalmag::Expr_literal Class Reference

```
#include <Expr_literal.h>
```

Inheritance diagram for genevalmag::Expr_literal:



Collaboration diagram for genevalmag::Expr_literal:



Public Member Functions

- `Expr_literal ()`
- `Expr_literal (const Expr_literal &other)`
- `virtual ~Expr_literal ()`
- `Expr_literal & operator= (const Expr_literal &other)`
- `literal_type get_type () const`
- `string get_value () const`
- `void set_type (literal_type new_type)`
- `void set_value (string new_value)`
- `string to_string () const`

Private Member Functions

- void `copy` (const `Expr_literal` &*other*)
- void `destroy` ()

Private Attributes

- string `value`
literal value.
- `literal_type` `type`
literal type: this type may be k_int, k_float, k_char, k_string.

7.16.1 Detailed Description

Definition at line 29 of file Expr_literal.h.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 genevalmag::Expr_literal::Expr_literal ()

Constructor empty of `Expr_literal`.

Returns

Constructor empty of `Expr_literal`.

Definition at line 23 of file Expr_literal.cpp.

7.16.2.2 genevalmag::Expr_literal::Expr_literal (const Expr_literal & *other*)

Constructor copy of `Expr_literal`.

Parameters

other

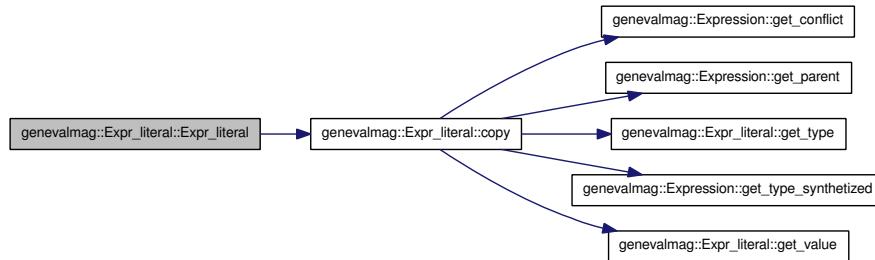
Returns

Constructor copy of `Expr_literal`.

Definition at line 33 of file Expr_literal.cpp.

References `copy()`.

Here is the call graph for this function:



7.16.2.3 `genevalmag::Expr_literal::~Expr_literal () [virtual]`

Destructor of [Expr_literal](#).

Returns

Destructor of [Expr_literal](#).

Definition at line 45 of file `Expr_literal.cpp`.

References `destroy()`.

Here is the call graph for this function:



7.16.3 Member Function Documentation

7.16.3.1 `void genevalmag::Expr_literal::copy (const Expr_literal & other) [private]`

Method of copy the [Expr_literal](#), STL-like C++.

Parameters

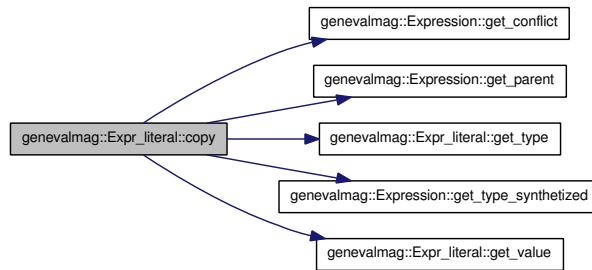
other Method of copy the [Expr_literal](#), STL-like C++.

Definition at line 74 of file `Expr_literal.cpp`.

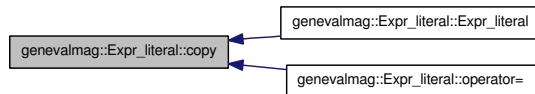
References `genevalmag::Expression::conflict`, `genevalmag::Expression::get_conflict()`, `genevalmag::Expression::get_parent()`, `get_type()`, `genevalmag::Expression::get_type_synthesized()`, `get_value()`, `genevalmag::Expression::parent`, `type`, `genevalmag::Expression::type_synthesized`, and `value`.

Referenced by `Expr_literal()`, and `operator=()`.

Here is the call graph for this function:



Here is the caller graph for this function:



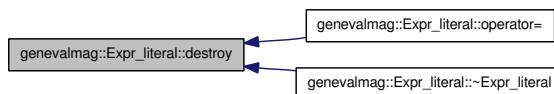
7.16.3.2 void genevalmag::Expr_literal::destroy () [private]

Method destroy [Expr_literal](#), STL-like C++.

Definition at line 86 of file Expr_literal.cpp.

Referenced by [operator=\(\)](#), and [~Expr_literal\(\)](#).

Here is the caller graph for this function:



7.16.3.3 literal_type genevalmag::Expr_literal::get_type () const

Returns the type of the [Expr_literal](#).

Returns

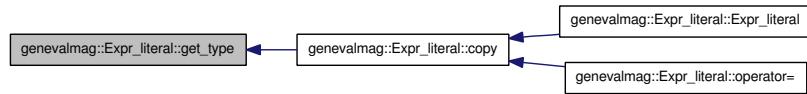
Returns the type of the [Expr_literal](#).

Definition at line 93 of file Expr_literal.cpp.

References [type](#).

Referenced by [copy\(\)](#).

Here is the caller graph for this function:



7.16.3.4 string genevalmag::Expr_literal::get_value () const

Returns the value of the [Expr_literal](#).

Returns

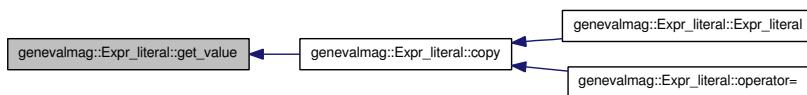
Returns the value of the [Expr_literal](#).

Definition at line 101 of file Expr_literal.cpp.

References value.

Referenced by copy0.

Here is the caller graph for this function:



7.16.3.5 Expr_literal & genevalmag::Expr_literal::operator= (const Expr_literal & other)

Operator assign(=) of [Expr_literal](#).

Parameters

other

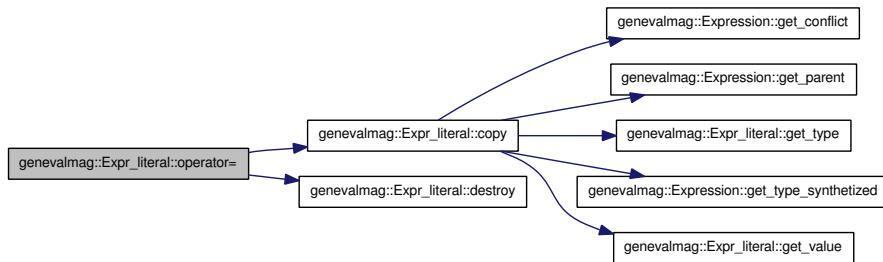
Returns

Operator assign(=) of [Expr_literal](#).

Definition at line 61 of file Expr_literal.cpp.

References copy0, and destroy0.

Here is the call graph for this function:



7.16.3.6 void genevalmag::Expr_literal::set_type (literal_type new_type)

Sets the type of the [Expr_literal](#).

Parameters

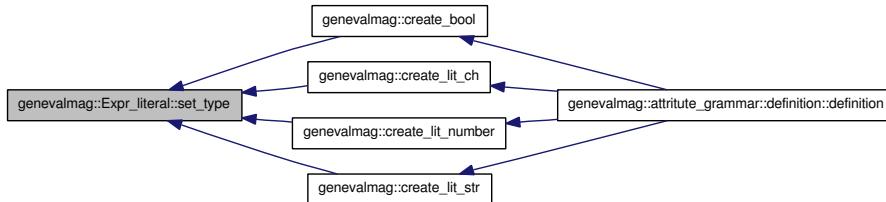
new_type Sets the type of the [Expr_literal](#).

Definition at line 109 of file Expr_literal.cpp.

References type.

Referenced by `genevalmag::create_bool()`, `genevalmag::create_lit_ch()`, `genevalmag::create_lit_number()`, and `genevalmag::create_lit_str()`.

Here is the caller graph for this function:



7.16.3.7 void genevalmag::Expr_literal::set_value (string new_value)

Sets the value of the [Expr_literal](#).

Parameters

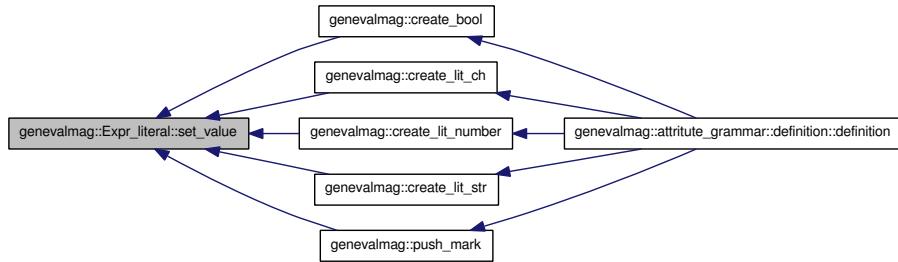
new_value Sets the value of the [Expr_literal](#).

Definition at line 117 of file Expr_literal.cpp.

References value.

Referenced by `genevalmag::create_bool()`, `genevalmag::create_lit_ch()`, `genevalmag::create_lit_number()`, `genevalmag::create_lit_str()`, and `genevalmag::push_mark()`.

Here is the caller graph for this function:



7.16.3.8 string genevalmag::Expr_literal::to_string () const [virtual]

Generate and return a string representation of a [Expr_literal](#).

Result= literal

Ex: 1 -> int, short or long

1.0 -> float or double

'1' -> char or wchar_t

"1" -> string

true -> bool

Returns

Generate and return a string representation of a [Expr_literal](#).

Result= literal

Ex: 1 -> int 1.0 -> float '1' -> char "1" -> string

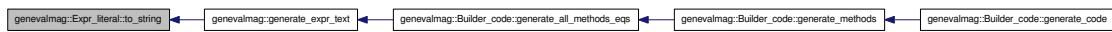
Implements [genevalmag::Expr_leaf](#).

Definition at line 132 of file Expr_literal.cpp.

References [genevalmag::k_bool](#), [genevalmag::k_char](#), [genevalmag::k_float](#), [genevalmag::k_int](#), [genevalmag::k_string](#), type, and value.

Referenced by [genevalmag::generate_expr_text\(\)](#).

Here is the caller graph for this function:



7.16.4 Member Data Documentation

7.16.4.1 [genevalmag::Expr_literal::type](#) [private]

literal type: this type may be k_int, k_float, k_char, k_string.

Definition at line 41 of file Expr_literal.h.

Referenced by copy(), get_type(), set_type(), and to_string().

7.16.4.2 genevalmag::Expr_literal::value [private]

literal value.

Definition at line 36 of file Expr_literal.h.

Referenced by copy(), get_value(), set_value(), and to_string().

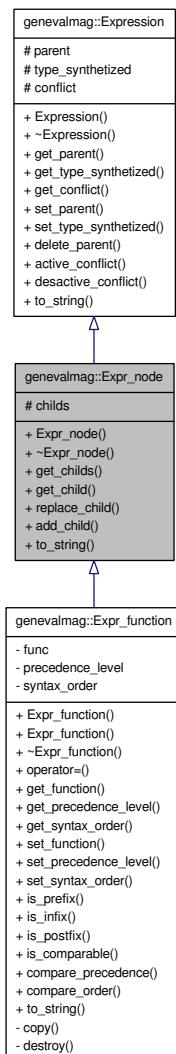
The documentation for this class was generated from the following files:

- include/Expression_tree/Expr_literal.h
- src/Expression_tree/Expr_literal.cpp

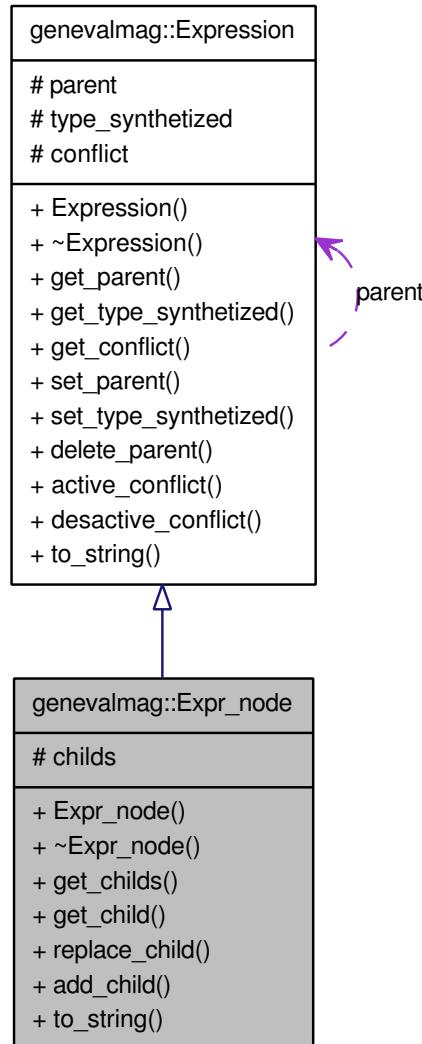
7.17 genevalmag::Expr_node Class Reference

```
#include <Expr_node.h>
```

Inheritance diagram for genevalmag::Expr_node:



Collaboration diagram for genevalmag::Expr_node:



Public Member Functions

- `Expr_node()`
- `virtual ~Expr_node()`
- `const vector< Expression * > & get_childs() const`
- `Expression * get_child(int index) const`
- `void replace_child(int index, Expression *other)`
- `void add_child(Expression *new_child)`
- `virtual string to_string() const =0`

Protected Attributes

- `vector< Expression * > child�`
Childs's vector of node.

7.17.1 Detailed Description

Definition at line 19 of file Expr_node.h.

7.17.2 Constructor & Destructor Documentation

7.17.2.1 `genevalmag::Expr_node::Expr_node ()`

Constructor of [Expr_node](#).

Returns

Definition at line 14 of file Expr_node.cpp.

7.17.2.2 `genevalmag::Expr_node::~Expr_node () [virtual]`

Destructor of [Expr_node](#).

Returns

Destructor of [Expr_node](#).

Definition at line 19 of file Expr_node.cpp.

References `child`.

7.17.3 Member Function Documentation

7.17.3.1 `void genevalmag::Expr_node::add_child (Expression * new_child)`

Adds a child in [Expr_function](#).

Updates the parent of the child.

Parameters

`new_child` Adds a child in [Expr_function](#). Updates the parent of the child.

Definition at line 57 of file Expr_node.cpp.

References `child`, and `genevalmag::Expression::set_parent()`.

Here is the call graph for this function:



7.17.3.2 Expression * genevalmag::Expr_node::get_child (int index) const

Returns the index-child of the `Expr_function`.

Parameters

`index`

Returns

Returns the index-child of the `Expr_function`.

Definition at line 38 of file `Expr_node.cpp`.

References `child`.

Referenced by `genevalmag::generate_expr_text()`.

Here is the caller graph for this function:



7.17.3.3 const vector< Expression * > & genevalmag::Expr_node::get_childs () const

Returns the vector with all children of the `Expr_function`.

Returns

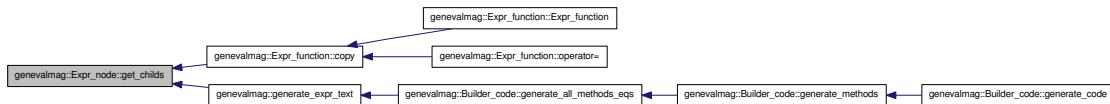
Returns the vector with all children of the `Expr_function`.

Definition at line 30 of file `Expr_node.cpp`.

References `child`.

Referenced by `genevalmag::Expr_function::copy()`, and `genevalmag::generate_expr_text()`.

Here is the caller graph for this function:



7.17.3.4 void genevalmag::Expr_node::replace_child (int index, Expression * other)

Replaces index-child of the `Expr_function` with `other`.

Updates the parent of the child.

Parameters

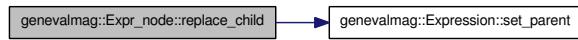
`index`

other Replaces index-child of the [Expr_function](#) with other. Updates the parent of the child.

Definition at line 47 of file Expr_node.cpp.

References `child`, and [genevalmag::Expression::set_parent\(\)](#).

Here is the call graph for this function:



7.17.3.5 virtual string [genevalmag::Expr_node::to_string\(\)](#) const [pure virtual]

Generate and return a string representation of an [Expr_node](#).

Returns

Implements [genevalmag::Expression](#).

Implemented in [genevalmag::Expr_function](#).

7.17.4 Member Data Documentation

7.17.4.1 [genevalmag::Expr_node::childs](#) [protected]

Childs's vector of node.

Definition at line 26 of file Expr_node.h.

Referenced by `add_child()`, [genevalmag::Expr_function::copy\(\)](#), `get_child()`, `get_childs()`, `replace_child()`, [genevalmag::Expr_function::to_string\(\)](#), and `~Expr_node()`.

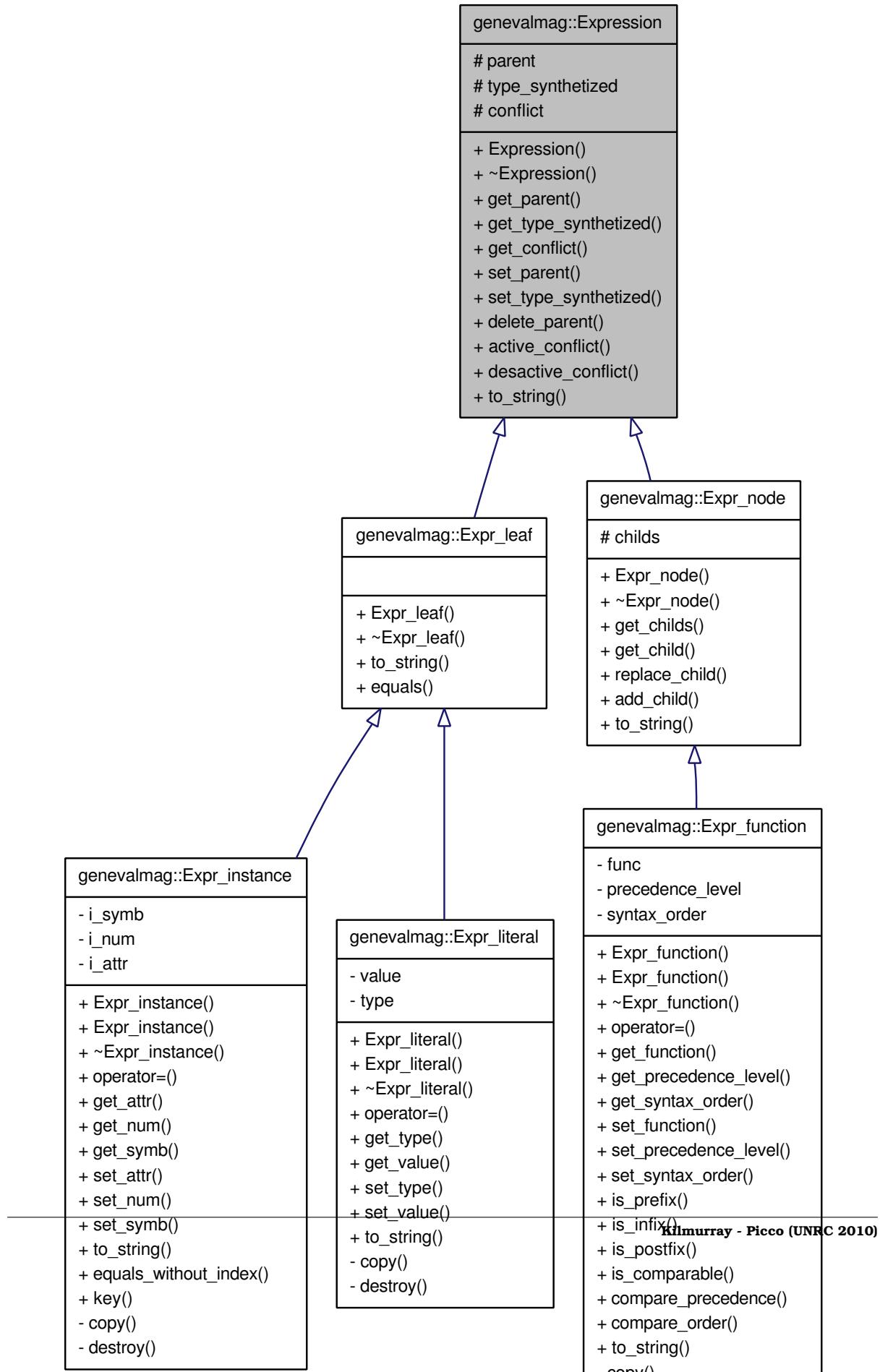
The documentation for this class was generated from the following files:

- include/Expression_tree/[Expr_node.h](#)
- src/Expression_tree/[Expr_node.cpp](#)

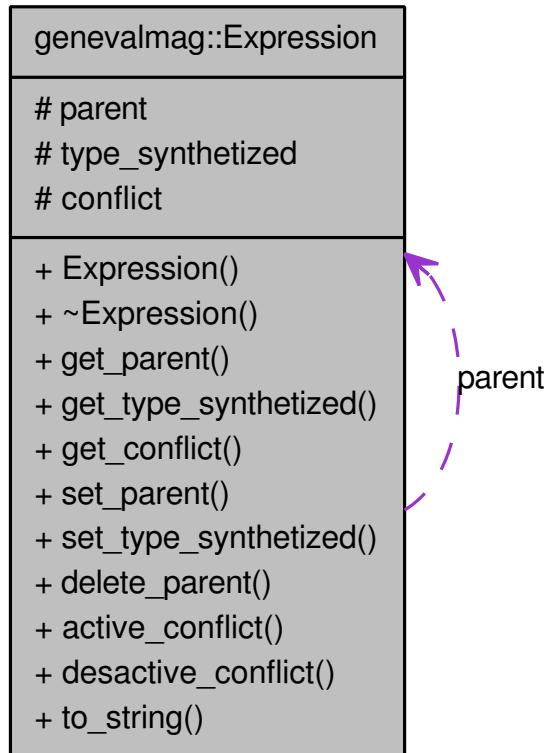
7.18 genevalmag::Expression Class Reference

```
#include <Expression.h>
```

Inheritance diagram for genevalmag::Expression:



Collaboration diagram for genevalmag::Expression:



Public Member Functions

- `Expression ()`
- `virtual ~Expression ()`
- `const Expression * get_parent () const`
- `const string get_type_synthesized () const`
- `int get_conflict () const`
- `void set_parent (const Expression *new_parent)`
- `void set_type_synthesized (string new_type_synthesized)`
- `void delete_parent ()`
- `void active_conflict (int prec_conflict)`
- `void desactive_conflict ()`
- `virtual string to_string () const =0`

Protected Attributes

- `const Expression * parent`
Father of Expression.
- `string type_synthesized`
Type of node: Obtains of parse.

- int **conflict**

*Mark when conflicted node.
This conflict is a conflict in the parse.
Used for define the precedence of function.*

7.18.1 Detailed Description

Definition at line 21 of file Expression.h.

7.18.2 Constructor & Destructor Documentation

7.18.2.1 `genevalmag::Expression::Expression ()`

Constructor of [Expression](#).

Returns

Constructor of [Expression](#).

Definition at line 17 of file Expression.cpp.

References conflict, parent, and type_synthesized.

7.18.2.2 `genevalmag::Expression::~Expression () [virtual]`

Destructor of [Expression](#).

Returns

Destructor of [Expression](#).

Definition at line 27 of file Expression.cpp.

7.18.3 Member Function Documentation

7.18.3.1 `void genevalmag::Expression::active_conflict (int prec_conflict)`

Turn on the conflict flag with the precedence wath produces.

Parameters

prec_conflict Turn on the conflict flag with the precedence wath produces.

Definition at line 80 of file Expression.cpp.

References conflict.

7.18.3.2 void genevalmag::Expression::delete_parent ()

Sets the parent pointer of the [Expression](#) in NULL.

Definition at line 72 of file Expression.cpp.

References parent.

7.18.3.3 void genevalmag::Expression::desactive_conflict ()

Turn off the conflict flag with negative precedence.

Definition at line 88 of file Expression.cpp.

References conflict.

7.18.3.4 int genevalmag::Expression::get_conflict () const

Returns the precedence wath produces the conflict.

Returns

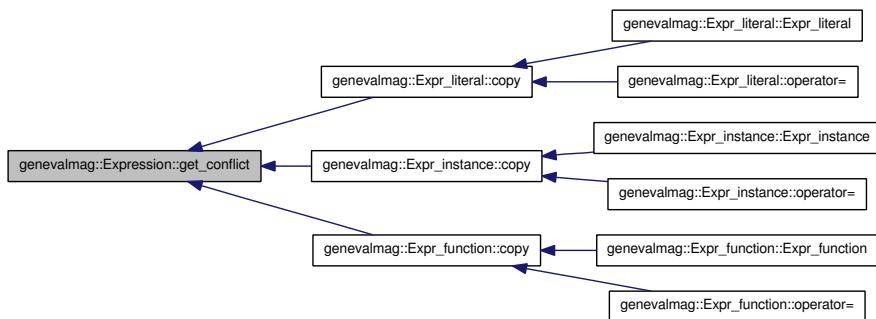
Returns the precedence wath produces the conflict.

Definition at line 48 of file Expression.cpp.

References conflict.

Referenced by `genevalmag::Expr_literal::copy()`, `genevalmag::Expr_instance::copy()`, and `genevalmag::Expr_function::copy()`.

Here is the caller graph for this function:

**7.18.3.5 const Expression * genevalmag::Expression::get_parent () const**

Returns the parent pointer of the [Expression](#).

Returns

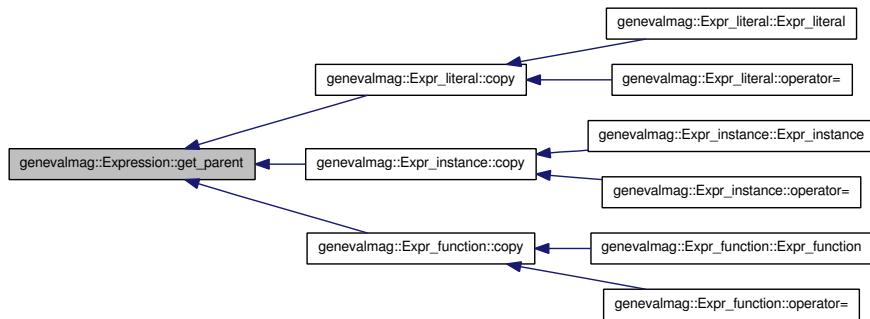
Returns the parent pointer of the [Expression](#).

Definition at line 32 of file Expression.cpp.

References parent.

Referenced by `genevalmag::Expr_literal::copy()`, `genevalmag::Expr_instance::copy()`, and `genevalmag::Expr_function::copy()`.

Here is the caller graph for this function:



7.18.3.6 const string genevalmag::Expression::get_type_synthesized () const

Returns the type synthesized of the [Expression](#).

Returns

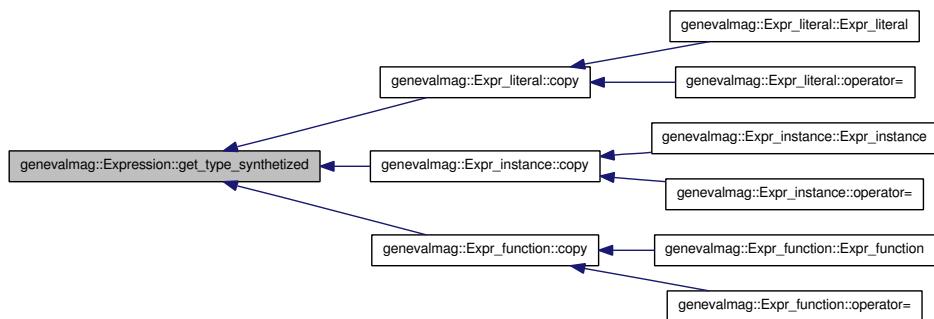
Returns the type synthesized of the [Expression](#).

Definition at line 40 of file Expression.cpp.

References type_synthesized.

Referenced by `genevalmag::Expr_literal::copy()`, `genevalmag::Expr_instance::copy()`, and `genevalmag::Expr_function::copy()`.

Here is the caller graph for this function:



7.18.3.7 void genevalmag::Expression::set_parent (const Expression * new_parent)

Sets the parent pointer of the Expression.

Parameters

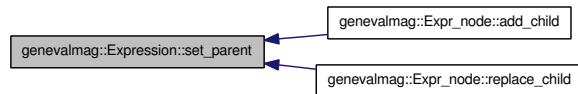
new_parent Sets the parent pointer of the Expression.

Definition at line 56 of file Expression.cpp.

References parent.

Referenced by genevalmag::Expr_node::add_child(), and genevalmag::Expr_node::replace_child().

Here is the caller graph for this function:



7.18.3.8 void genevalmag::Expression::set_type_synthesized (string new_type_synthesized)

Returns the type synthesized of the Expression.

Parameters

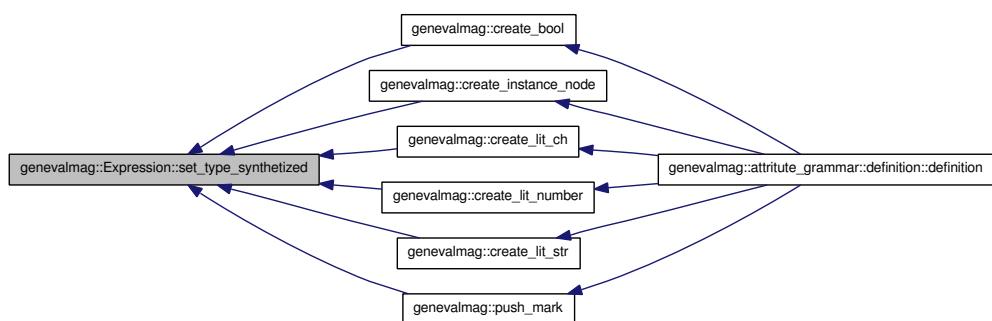
new_type_synthesized Returns the type synthesized of the Expression.

Definition at line 64 of file Expression.cpp.

References type_synthesized.

Referenced by genevalmag::create_bool(), genevalmag::create_instance_node(), genevalmag::create_lit_ch(), genevalmag::create_lit_number(), genevalmag::create_lit_str(), and genevalmag::push_mark().

Here is the caller graph for this function:



7.18.3.9 virtual string genevalmag::Expression::to_string () const [pure virtual]

Generate and return a string representation of a [Expression](#).

Returns

Implemented in [genevalmag::Expr_function](#), [genevalmag::Expr_instance](#), [genevalmag::Expr_leaf](#), [genevalmag::Expr_literal](#), and [genevalmag::Expr_node](#).

7.18.4 Member Data Documentation

7.18.4.1 genevalmag::Expression::conflict [protected]

Mark when conflicted node.

This conflict is a conflict in the parse.

Used for define the precedence of function.

.

Definition at line 40 of file Expression.h.

Referenced by [active_conflict\(\)](#), [genevalmag::Expr_literal::copy\(\)](#), [genevalmag::Expr_instance::copy\(\)](#), [genevalmag::Expr_function::copy\(\)](#), [desactive_conflict\(\)](#), [Expression\(\)](#), and [get_conflict\(\)](#).

7.18.4.2 genevalmag::Expression::parent [protected]

Father of [Expression](#).

Definition at line 28 of file Expression.h.

Referenced by [genevalmag::Expr_literal::copy\(\)](#), [genevalmag::Expr_instance::copy\(\)](#), [genevalmag::Expr_function::copy\(\)](#), [delete_parent\(\)](#), [Expression\(\)](#), [get_parent\(\)](#), and [set_parent\(\)](#).

7.18.4.3 genevalmag::Expression::type_synthesized [protected]

Type of node: Obtains of parse.

Definition at line 33 of file Expression.h.

Referenced by [genevalmag::Expr_literal::copy\(\)](#), [genevalmag::Expr_instance::copy\(\)](#), [genevalmag::Expr_function::copy\(\)](#), [Expression\(\)](#), [get_type_synthesized\(\)](#), and [set_type_synthesized\(\)](#).

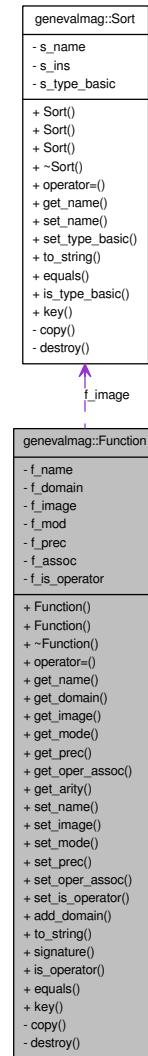
The documentation for this class was generated from the following files:

- [include/Expression_tree/Expression.h](#)
- [src/Expression_tree/Expression.cpp](#)

7.19 genevalmag::Function Class Reference

```
#include <Function.h>
```

Collaboration diagram for genevalmag::Function:



Public Member Functions

- `Function ()`
- `Function (const Function &other)`
- `virtual ~Function ()`
- `Function & operator= (const Function &other)`
- `string get_name () const`
- `const vector< const Sort * > & get_domain () const`
- `const Sort * get_image () const`
- `const oper_mode get_mode () const`
- `const unsigned short get_prec () const`

- const oper_assoc get_oper_assoc () const
- const int get_arity () const
- void set_name (const string name)
- void set_image (const Sort *image)
- void set_mode (const string mode)
- void set_prec (const unsigned short prec)
- void set_oper_assoc (const string mod)
- void set_is_operator (const bool value)
- void add_domain (const Sort *sort)
- const string to_string () const
- const string signature () const
- const bool is_operator () const
- const bool equals (const Function &other) const
- const string key () const

Private Member Functions

- void copy (const Function &other)
- void destroy ()

Private Attributes

- string f_name
Function's name.
- vector< const Sort * > f_domain
Function's sort domain.
- const Sort * f_image
Function's sort image.
- oper_mode f_mod
Function's notation. Functions are always prefix. Operators can be anyone.
- unsigned short f_prec
Function's precedence.
- oper_assoc f_assoc
Function's associativity.
- bool f_is_operator
If that value is TRUE then the object function is a operator. This can be: prefix, infix or postfix.

7.19.1 Detailed Description

Definition at line 40 of file Function.h.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 genevalmag::Function::Function ()

Constructor empty of function.

Constructor empty of [Function](#).

Definition at line 21 of file Function.cpp.

References f_assoc, f_is_operator, f_mod, f_prec, genevalmag::k_left, and genevalmag::k_-prefix.

7.19.2.2 genevalmag::Function::Function (const Function & other)

Constructor copy of function.

Parameters

other

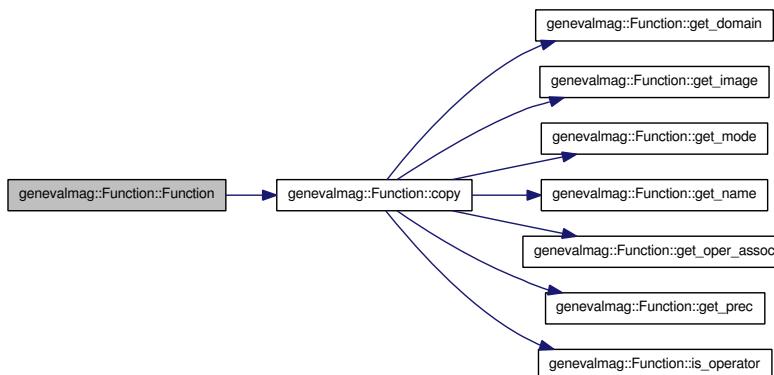
Returns

Constructor copy of [Function](#).

Definition at line 32 of file Function.cpp.

References copy0().

Here is the call graph for this function:



7.19.2.3 genevalmag::Function::~Function () [virtual]

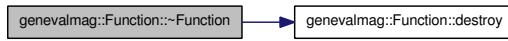
Destructor of the function.

Destructor of the [Function](#).

Definition at line 40 of file Function.cpp.

References destroy().

Here is the call graph for this function:



7.19.3 Member Function Documentation

7.19.3.1 void genevalmag::Function::add_domain (const Sort * sort)

Enqueues a sort in the domain of the **Function**.

Parameters

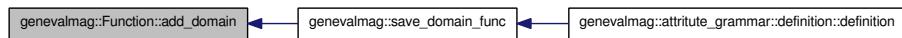
sort Enqueue a sort in the domain of the **Function**.

Definition at line 230 of file Function.cpp.

References f_domain.

Referenced by genevalmag::save_domain_func().

Here is the caller graph for this function:



7.19.3.2 void genevalmag::Function::copy (const Function & other) [private]

Method of copy the function, STL-like C++.

Parameters

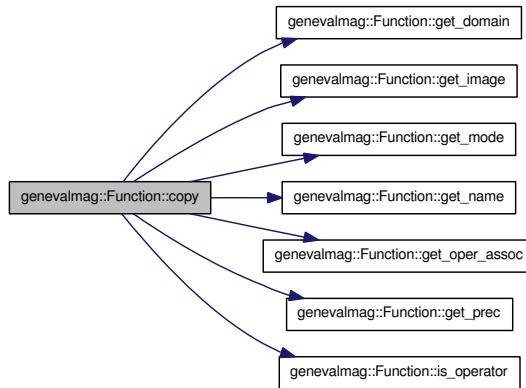
other Method of copy the **Function**, STL-like C++.

Definition at line 61 of file Function.cpp.

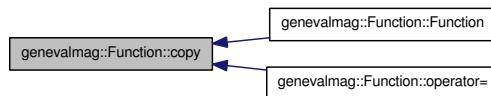
References f_assoc, f_domain, f_image, f_is_operator, f_mod, f_name, f_prec, get_domain(), get_image(), get_mode(), get_name(), get_oper_assoc(), get_prec(), and is_operator().

Referenced by Function(), and operator=().

Here is the call graph for this function:



Here is the caller graph for this function:



7.19.3.3 void genevalmag::Function::destroy () [private]

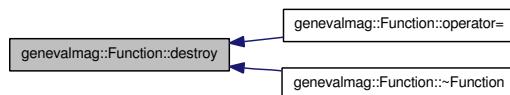
Method destroy function, STL-like C++.

Method destroy [Function](#), STL-like C++.

Definition at line 75 of file Function.cpp.

Referenced by [operator=\(\)](#), and [~Function\(\)](#).

Here is the caller graph for this function:



7.19.3.4 const bool genevalmag::Function::equals (const Function & other) const

Compares the [Function](#) with other.

Parameters

other

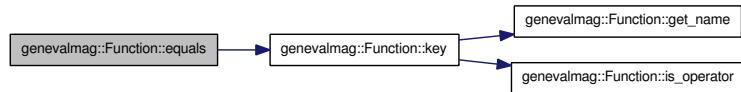
Returns

Compares the [Function](#) with other.

Definition at line 346 of file Function.cpp.

References key().

Here is the call graph for this function:



7.19.3.5 const int genevalmag::Function::get_arity () const

Returns the arity of Function.

Returns

Returns the arity of Function.

Definition at line 130 of file Function.cpp.

References f_domain.

7.19.3.6 const vector< const Sort * > & genevalmag::Function::get_domain () const

Returns the domain of the [Function](#).

Returns

Returns the domain of the [Function](#).

Definition at line 90 of file Function.cpp.

References f_domain.

Referenced by copy().

Here is the caller graph for this function:



7.19.3.7 const Sort * genevalmag::Function::get_image () const

Returns the image of the [Function](#).

Returns

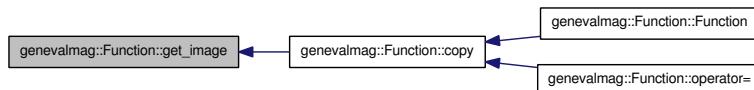
Returns the image of the [Function](#).

Definition at line 98 of file Function.cpp.

References f_image.

Referenced by copy().

Here is the caller graph for this function:

**7.19.3.8 const oper_mode genevalmag::Function::get_mode () const**

Returns the mode of the operator.

Returns

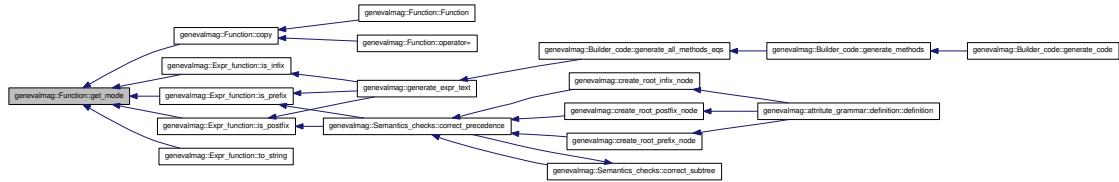
Returns the mode of the operator.

Definition at line 106 of file Function.cpp.

References f_mod.

Referenced by copy(), genevalmag::Expr_function::is_infix(), genevalmag::Expr_function::is_postfix(), genevalmag::Expr_function::is_prefix(), and genevalmag::Expr_function::to_string().

Here is the caller graph for this function:

**7.19.3.9 string genevalmag::Function::get_name () const**

Returns the name of the [Function](#).

Returns

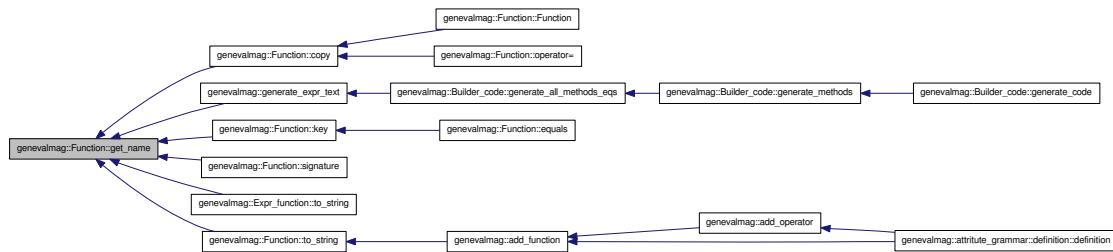
Returns the name of the [Function](#).

Definition at line 82 of file Function.cpp.

References f_name.

Referenced by copy(), genevalmag::generate_expr_text(), key(), signature(), genevalmag::Expr_function::to_string(), and to_string().

Here is the caller graph for this function:



7.19.3.10 const oper_assoc genevalmag::Function::get_oper_assoc () const

Returns the associativity of the operator.

Returns

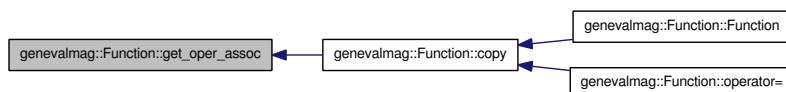
Returns the associativity of the operator.

Definition at line 122 of file Function.cpp.

References f_assoc.

Referenced by copy().

Here is the caller graph for this function:



7.19.3.11 const unsigned short genevalmag::Function::get_prec () const

Returns the precedence of the operator.

Returns

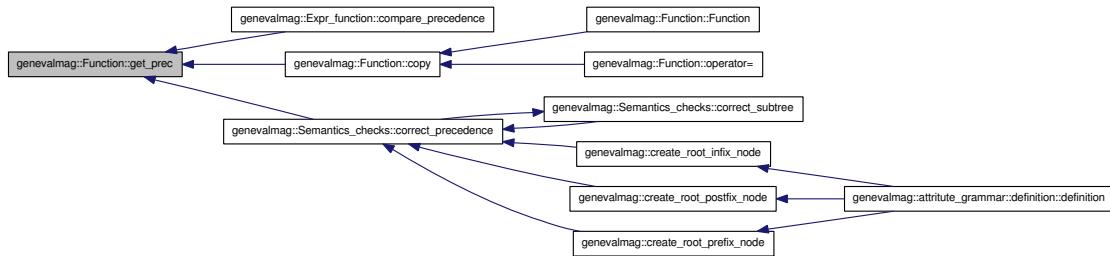
Returns the precedence of the operator.

Definition at line 114 of file Function.cpp.

References f_prec.

Referenced by genevalmag::Expr_function::compare_precedence(), copy0, and genevalmag::Semantics_checks::correct_precedence().

Here is the caller graph for this function:



7.19.3.12 const bool genevalmag::Function::is_operator () const

Returns true if the Function is a Operator.

Returns

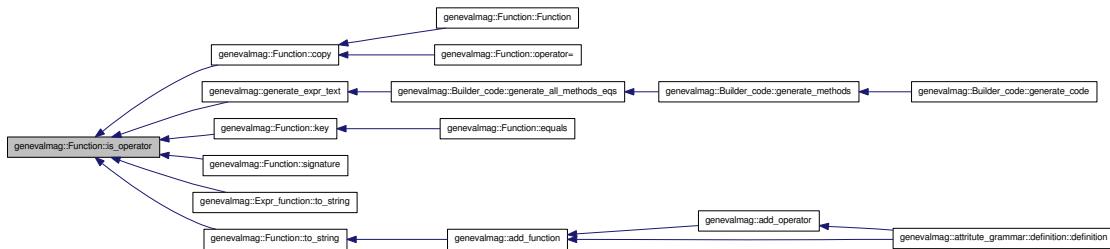
Returns true if the Function is a Operator.

Definition at line 338 of file Function.cpp.

References f_is_operator.

Referenced by copy0, genevalmag::generate_expr_text(), key0, signature0, genevalmag::Expr_function::to_string(), and to_string0.

Here is the caller graph for this function:



7.19.3.13 const string genevalmag::Function::key () const

Generates and returns the string key that identifies a Function definitely.

Result = <name> <domain> <image>

where <domain> is = sort_1 ... sort_n

Returns

Generates and returns the string key that identifies a [Function](#) definitely.

Result = <name> <domain> <image>

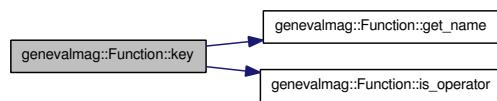
where <domain> is = sort_1 ... sort_n

Definition at line 358 of file Function.cpp.

References `f_domain`, `f_mod`, `f_name`, `get_name()`, `is_operator()`, `genevalmag::k_infix`, `genevalmag::k_postfix`, and `genevalmag::k_prefix`.

Referenced by `equals()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.19.3.14 Function & genevalmag::Function::operator= (const Function & other)

Operator assign(=) of function.

Parameters

other

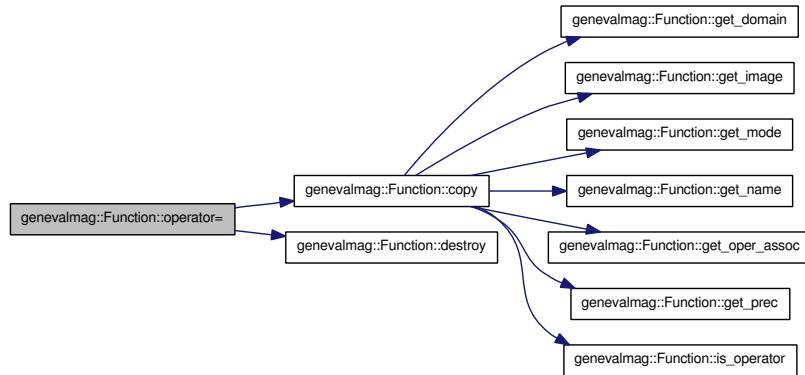
Returns

Operator assign(=) of [Function](#).

Definition at line 48 of file Function.cpp.

References `copy()`, and `destroy()`.

Here is the call graph for this function:



7.19.3.15 void genevalmag::Function::set_image (const Sort * *image*)

Sets the sort image of the [Function](#).

Parameters

image Sets the sort image of the [Function](#).

Definition at line 146 of file Function.cpp.

References f_image.

Referenced by genevalmag::save_image_func().

Here is the caller graph for this function:



7.19.3.16 void genevalmag::Function::set_is_operator (const bool *value*)

Sets the boolean attribute with the parameter.

Parameters

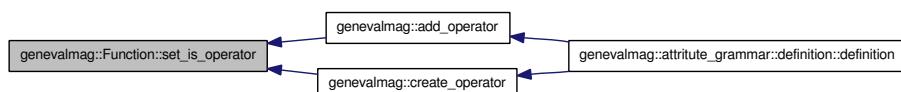
value Sets the boolean attribute with the parameter.

Definition at line 222 of file Function.cpp.

References f_is_operator.

Referenced by genevalmag::add_operator(), and genevalmag::create_operator().

Here is the caller graph for this function:



7.19.3.17 void genevalmag::Function::set_mode (const string mode)

Sets the mode of the operator.

Parameters

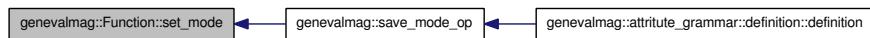
mode Sets the mode of the operator.

Definition at line 154 of file Function.cpp.

References f_mod, genevalmag::k_infix, genevalmag::k_postfix, and genevalmag::k_prefix.

Referenced by genevalmag::save_mode_op().

Here is the caller graph for this function:



7.19.3.18 void genevalmag::Function::set_name (const string name)

Sets the name of the Function.

Parameters

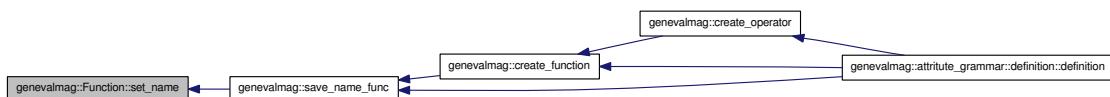
name Sets the name of the Function.

Definition at line 138 of file Function.cpp.

References f_name.

Referenced by genevalmag::save_name_func().

Here is the caller graph for this function:



7.19.3.19 void genevalmag::Function::set_oper_assoc (const string mod)

Sets the associativity of the operator.

Parameters

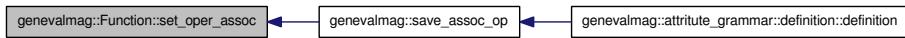
mod Sets the associativity of the operator.

Definition at line 192 of file Function.cpp.

References f_assoc, genevalmag::k_left, genevalmag::k_non_assoc, and genevalmag::k_right.

Referenced by genevalmag::save_assoc_op().

Here is the caller graph for this function:



7.19.3.20 void genevalmag::Function::set_prec (const unsigned short prec)

Sets the precedence of the operator.

Parameters

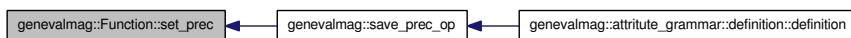
prec Sets the precedence of the operator.

Definition at line 184 of file Function.cpp.

References f_prec.

Referenced by genevalmag::save_prec_op().

Here is the caller graph for this function:



7.19.3.21 const string genevalmag::Function::signature () const

Generates and returns a string with the signature of a [Function](#).

Result = <image> <name> "(" <domain> ");"

or if is an operator:

Result = <image> "operator"<name> "(" <domain> ");"

where <domain> is = sort_1 P1", " ... ", " sort_n Pn

Returns

Generates and returns a string with the signature of a [Function](#).

Result = <image> <name> "(" <domain> ");"

or if is an operator:

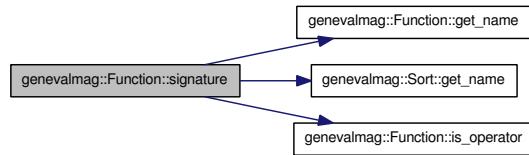
Result = <image> "operator"<name> "(" <domain> ");"

where <domain> is= sort_1 P1", " ... ", " sort_n Pn

Definition at line 308 of file Function.cpp.

References f_domain, f_image, f_name, get_name(), genevalmag::Sort::get_name(), and is_-operator().

Here is the call graph for this function:



7.19.3.22 const string genevalmag::Function::to_string () const

Generates and returns a string representation of a [Function](#).

Result = "function" <name> ":" <domain> "->" <image> ";"

where <domain> is= sort_1 ["(" <instance> ")" IF DEBUG IS ON] "," ... "," sort_n ["(" <instance> ")" IF DEBUG IS ON]

Returns

Generates and returns a string representation of a [Function](#).

Result = "function" <name> ":" <domain> "->" <image> ";"

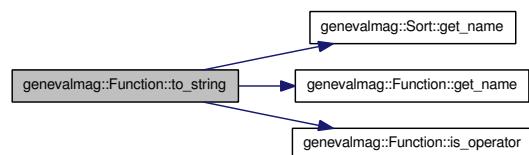
where <domain> is= sort_1 ["(" <instance> ")" IF DEBUG IS ON] "," ... "," sort_n ["(" <instance> ")" IF DEBUG IS ON]

Definition at line 242 of file Function.cpp.

References f_assoc, f_domain, f_image, f_mod, f_name, f_prec, genevalmag::Sort::get_name(), get_name(), is_operator(), genevalmag::k_infix, genevalmag::k_left, genevalmag::k_non_assoc, genevalmag::k_postfix, genevalmag::k_prefix, and genevalmag::k_right.

Referenced by genevalmag::add_function().

Here is the call graph for this function:



Here is the caller graph for this function:



7.19.4 Member Data Documentation

7.19.4.1 genevalmag::Function::f_assoc [private]

Function's associativity.

Definition at line 72 of file Function.h.

Referenced by copy(), Function(), get_oper_assoc(), set_oper_assoc(), and to_string().

7.19.4.2 genevalmag::Function::f_domain [private]

Function's sort domain.

Definition at line 52 of file Function.h.

Referenced by add_domain(), copy(), get_arity(), get_domain(), key(), signature(), and to_string().

7.19.4.3 genevalmag::Function::f_image [private]

Function's sort image.

Definition at line 57 of file Function.h.

Referenced by copy(), get_image(), set_image(), signature(), and to_string().

7.19.4.4 genevalmag::Function::f_is_operator [private]

If that value is TRUE then the object function is a operator. This can be: prefix, infix or postfix.

Definition at line 78 of file Function.h.

Referenced by copy(), Function(), is_operator(), and set_is_operator().

7.19.4.5 genevalmag::Function::f_mod [private]

Function's notation. Functions are always prefix. Operators can be anyone.

Definition at line 62 of file Function.h.

Referenced by copy(), Function(), get_mode(), key(), set_mode(), and to_string().

7.19.4.6 genevalmag::Function::f_name [private]

Function's name.

Definition at line 47 of file Function.h.

Referenced by copy(), get_name(), key(), set_name(), signature(), and to_string().

7.19.4.7 genevalmag::Function::f_prec [private]

Function's precedence.

Definition at line 67 of file Function.h.

Referenced by copy(), Function(), get_prec(), set_prec(), and to_string().

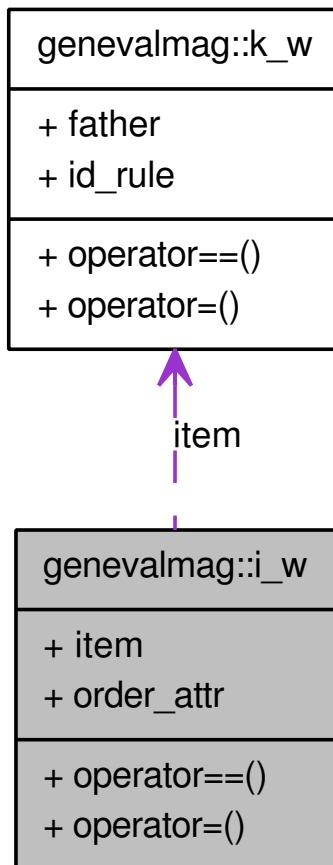
The documentation for this class was generated from the following files:

- include/Attr_grammar/Function.h
- src/Attr_grammar/Function.cpp

7.20 genevalmag::i_w Struct Reference

```
#include <Builder_plans.h>
```

Collaboration diagram for genevalmag::i_w:



Public Member Functions

- bool `operator== (const i_w &other) const`
- `i_w & operator= (const i_w &other)`

Public Attributes

- `Key_work_list item`
A key_work_list.
- `unsigned short order_attr`
Order of equation.

7.20.1 Detailed Description

This struct represents an Item work, that is an Key_work_list and an equations order evaluation.

Definition at line 114 of file Builder_plans.h.

7.20.2 Member Function Documentation

7.20.2.1 `i_w& genevalmag::i_w::operator= (const i_w & other) [inline]`

operator=

Parameters

other

Returns

Definition at line 140 of file Builder_plans.h.

References item, and order_attr.

7.20.2.2 `bool genevalmag::i_w::operator== (const i_w & other) const [inline]`

operator==

Parameters

other

Returns

Definition at line 131 of file Builder_plans.h.

References item, and order_attr.

7.20.3 Member Data Documentation

7.20.3.1 `genevalmag::i_w::item`

A key_work_list.

Definition at line 120 of file Builder_plans.h.

Referenced by genevalmag::Builder_plans::generate_plans(), operator=(), and operator==().

7.20.3.2 `genevalmag::i_w::order_attr`

Order of ecuation.

Definition at line 125 of file Builder_plans.h.

Referenced by genevalmag::Builder_plans::generate_plans(), operator=(), and operator==().

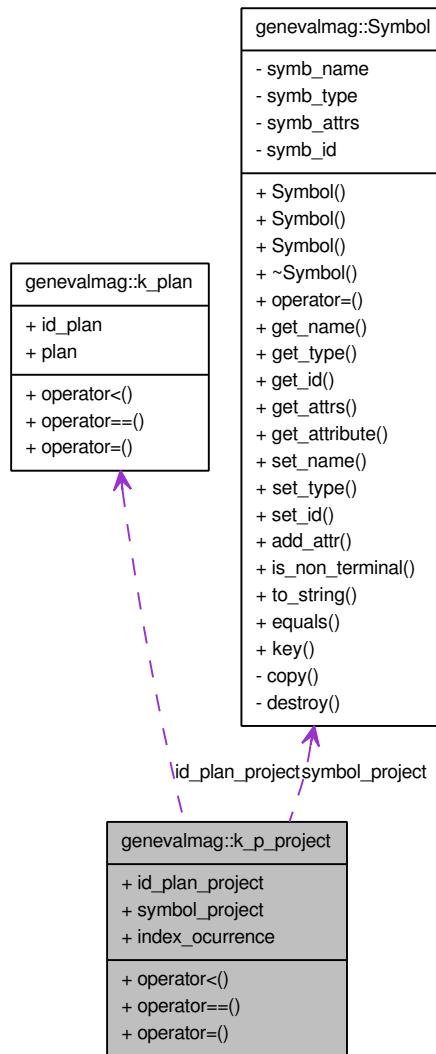
The documentation for this struct was generated from the following file:

- include/Builders/Builder_plans.h

7.21 genevalmag::k_p_project Struct Reference

```
#include <Builder_plans.h>
```

Collaboration diagram for genevalmag::k_p_project:



Public Member Functions

- `bool operator< (const k_p_project &other) const`
- `bool operator== (const k_p_project &other) const`
- `k_p_project & operator= (const k_p_project &other)`

Public Attributes

- `Key_plan id_plan_project`
key_plan

- const Symbol * symbol_project
Projected Symbol.
- unsigned short index_ocurrence

7.21.1 Detailed Description

This struct represent a Key_plan_project, that is a Key_plan, the symbol and occurrence, by which project.

Definition at line 200 of file Builder_plans.h.

7.21.2 Member Function Documentation

7.21.2.1 `bool genevalmag::k_p_project::operator< (const k_p_project & other) const [inline]`

operator<

Parameters

other

Returns

Definition at line 222 of file Builder_plans.h.

References genevalmag::Symbol::get_name(), id_plan_project, index_ocurrence, and symbol_project.

Here is the call graph for this function:



7.21.2.2 `k_p_project& genevalmag::k_p_project::operator= (const k_p_project & other) [inline]`

operator=

Parameters

other

Returns

Definition at line 242 of file Builder_plans.h.

References id_plan_project, index_ocurrence, and symbol_project.

7.21.2.3 **bool genevalmag::k_p_project::operator== (const k_p_project & other) const [inline]**

operator==

Parameters

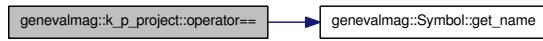
other

Returns

Definition at line 233 of file Builder_plans.h.

References genevalmag::Symbol::get_name(), id_plan_project, index_ocurrence, and symbol_project.

Here is the call graph for this function:



7.21.3 Member Data Documentation

7.21.3.1 **genevalmag::k_p_project::id_plan_project**

key_plan

Definition at line 206 of file Builder_plans.h.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), genevalmag::Builder_plans::generate_plans(), operator<(), operator=(), and operator==().

7.21.3.2 **unsigned short genevalmag::k_p_project::index_ocurrence**

Definition at line 216 of file Builder_plans.h.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), genevalmag::Builder_plans::generate_plans(), operator<(), operator=(), and operator==().

7.21.3.3 **genevalmag::k_p_project::symbol_project**

Projected Symbol.

Definition at line 211 of file Builder_plans.h.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), genevalmag::Builder_plans::generate_plans(), operator<(), operator=(), and operator==().

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

- include/Builders/Builder_plans.h

7.22 genevalmag::k_plan Struct Reference

```
#include <Builder_plans.h>
```

Public Member Functions

- bool `operator< (const k_plan &other) const`
- bool `operator== (const k_plan &other) const`
- `k_plan & operator= (const k_plan &other)`

Public Attributes

- unsigned short `id_plan`
Index plan. - index to order superior.
- unsigned short `plan`
Index plan.

7.22.1 Detailed Description

This structs represent a Key_plan, that is an id and an equations order evaluation.
Definition at line 152 of file Builder_plans.h.

7.22.2 Member Function Documentation

7.22.2.1 `bool genevalmag::k_plan::operator< (const k_plan & other) const [inline]`

`operator<`

Parameters

other

Returns

Definition at line 169 of file Builder_plans.h.

References `id_plan`, and `plan`.

7.22.2.2 `k_plan& genevalmag::k_plan::operator= (const k_plan & other) [inline]`

`operator=`

Parameters

other

Returns

Definition at line 188 of file Builder_plans.h.

References id_plan, and plan.

7.22.2.3 `bool genevalmag::k_plan::operator== (const k_plan & other) const [inline]`
operator==

Parameters

other

Returns

Definition at line 179 of file Builder_plans.h.

References id_plan, and plan.

7.22.3 Member Data Documentation

7.22.3.1 `genevalmag::k_plan::id_plan`

Index plan. - index to order superior.

Definition at line 158 of file Builder_plans.h.

Referenced by genevalmag::generate_key_plan(), genevalmag::Builder_plans::generate_-
plans(), operator<(), operator=(), and operator==().

7.22.3.2 `genevalmag::k_plan::plan`

Index plan.

Definition at line 163 of file Builder_plans.h.

Referenced by genevalmag::generate_key_plan(), genevalmag::Builder_plans::generate_-
plans(), operator<(), operator=(), and operator==().

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

- include/Builders/Builder_plans.h

7.23 genevalmag::k_w Struct Reference

```
#include <Builder_plans.h>
```

Public Member Functions

- bool `operator== (const k_w &other) const`
- `k_w & operator= (const k_w &other)`

Public Attributes

- unsigned short `father`
Index context.father (rule).
- unsigned short `id_rule`
The index of rule.

7.23.1 Detailed Description

This struct represents a Key_work_list, that is a father rule and id-rule of the current rule.
Definition at line 76 of file Builder_plans.h.

7.23.2 Member Function Documentation

7.23.2.1 `k_w& genevalmag::k_w::operator= (const k_w & other) [inline]`

`operator=`

Parameters

other

Returns

Definition at line 102 of file Builder_plans.h.

References father, and id_rule.

7.23.2.2 `bool genevalmag::k_w::operator== (const k_w & other) const [inline]`

`operator==`

Parameters

other

Returns

Definition at line 93 of file Builder_plans.h.

References father, and id_rule.

7.23.3 Member Data Documentation

7.23.3.1 genevalmag::k_w::father

Index context father (rule).

Definition at line 82 of file Builder_plans.h.

Referenced by genevalmag::Builder_plans::generate_plans(), operator=(), and operator==().

7.23.3.2 genevalmag::k_w::id_rule

The index of rule.

Definition at line 87 of file Builder_plans.h.

Referenced by genevalmag::Builder_plans::generate_plans(), operator=(), and operator==().

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

- include/Builders/Builder_plans.h

7.24 genevalmag::Maglib Class Reference

```
#include <Maglib.h>
```

Public Member Functions

- `Maglib ()`
- `virtual ~Maglib ()`
- `int gen_evaluator (const string path_input_file, const string path_folder_output, const string name_library, const vector< string > &headers, const string path) const`

7.24.1 Detailed Description

Definition at line 19 of file Maglib.h.

7.24.2 Constructor & Destructor Documentation

7.24.2.1 genevalmag::Maglib::Maglib ()

Constructor empty of Mag_lib.

Definition at line 26 of file Maglib.cpp.

7.24.2.2 genevalmag::Maglib::~Maglib () [virtual]

Destructor empty of Mag_lib.

Definition at line 33 of file Maglib.cpp.

7.24.3 Member Function Documentation

7.24.3.1 int genevalmag::Maglib::gen_evaluator (const string path_input_file, const string path_folder_output, const string name_library, const vector< string > & headers, const string path) const

Generates the static evaluator's files for the `Attribute` grammar specified in the input_path_file passed as parameter. All outputs of the library be saved in the path_folder_output. The evaluator classname is defined by name_library. The user could include somes headers for uses externs functions to evaluator.

Parameters

`path_input_file`
`path_folder_output`
`name_library`
`headers`
`path`

Returns

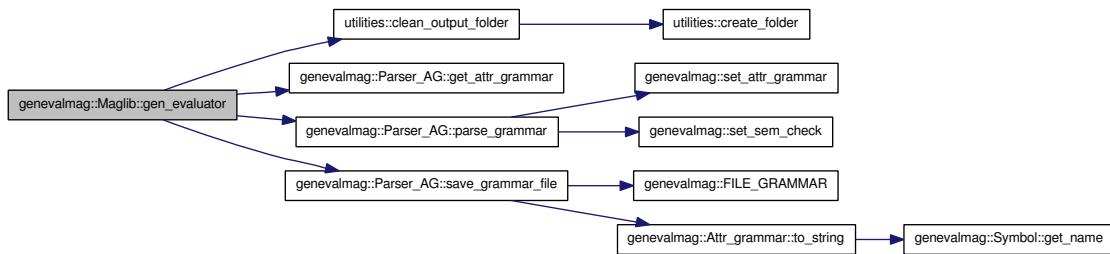
Generates the static evaluator's files for the [Attribute](#) grammar specified in the input_path_file passed as parameter. All outputs of the library be saved in the path_folder_output. The evaluator classname is defined by name_library. The user could include somes headers for uses externs functions to evaluator.

Definition at line 45 of file Maglib.cpp.

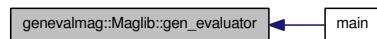
References `utilities::clean_output_folder()`, `genevalmag::Parser_AG::get_attr_grammar()`, `genevalmag::Parser_AG::parse_grammar()`, and `genevalmag::Parser_AG::save_grammar_file()`.

Referenced by `main()`.

Here is the call graph for this function:



Here is the caller graph for this function:



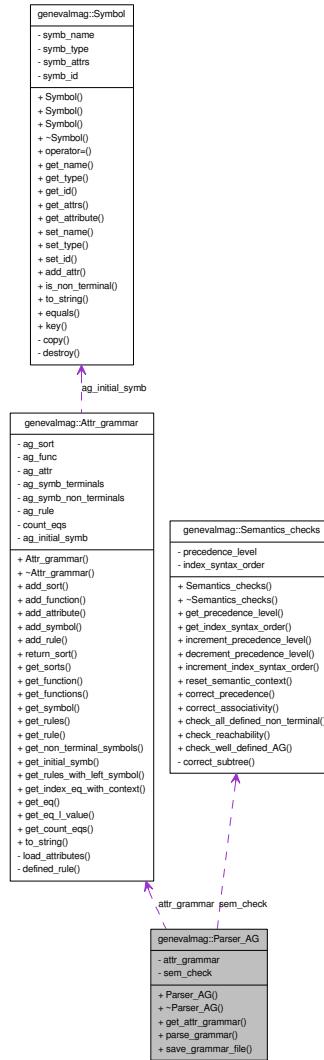
The documentation for this class was generated from the following files:

- [include/Maglib.h](#)
- [src/Maglib.cpp](#)

7.25 genevalmag::Parser_AG Class Reference

```
#include <Parser_AG.h>
```

Collaboration diagram for genevalmag::Parser_AG:



Public Member Functions

- `Parser_AG()`
- `~Parser_AG()`
- `const Attr_grammar & get_attr_grammar()`
- `bool parse_grammar(const string &path_file_input)`
- `void save_grammar_file(const string path_output) const`

Private Attributes

- `Attr_grammar attr_grammar`

Internal structure of grammar.

- [Semantics_checks sem_check](#)

Semantics control for grammar: Well define grammar.

7.25.1 Detailed Description

Definition at line 32 of file Parser_AG.h.

7.25.2 Constructor & Destructor Documentation

7.25.2.1 [genevalmag::Parser_AG::Parser_AG \(\)](#)

Constructor of [Parser_AG](#).

Definition at line 33 of file Parser_AG.cpp.

7.25.2.2 [genevalmag::Parser_AG::~Parser_AG \(\)](#)

Destructor of [Parser_AG](#).

Definition at line 39 of file Parser_AG.cpp.

7.25.3 Member Function Documentation

7.25.3.1 [const Attr_grammar & genevalmag::Parser_AG::get_attr_grammar \(\)](#)

Returns att_grammar of parser.

Returns

Returns att_grammar of parser.

Definition at line 46 of file Parser_AG.cpp.

References attr_grammar.

Referenced by [genevalmag::Maglib::gen_evaluator\(\)](#).

Here is the caller graph for this function:



7.25.3.2 [bool genevalmag::Parser_AG::parse_grammar \(const string & path_file_input\)](#)

This method invokes the method 'parse' of the library Spirit included in Boost. Returns true if could parse all the input.

Parameters

path_file_input

Returns

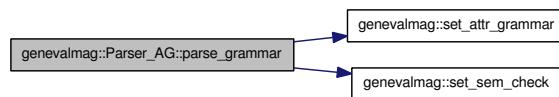
This method invokes the method 'parse' of the library Spitir included in Boost. Returns true if could parse all the input.

Definition at line 399 of file Parser_AG.cpp.

References attr_grammar, sem_check, genevalmag::set_attr_grammar(), and genevalmag::set_sem_check().

Referenced by genevalmag::Maglib::gen_evaluator().

Here is the call graph for this function:



Here is the caller graph for this function:



7.25.3.3 void genevalmag::Parser_AG::save_grammar_file (const string path_output) const

This method saves all parsed string in a file. The file output is in argument path_output.

Parameters

path_output This method saves all parsed string in a file. The file output is in argument path_output.

Definition at line 447 of file Parser_AG.cpp.

References attr_grammar, genevalmag::FILE_GRAMMAR(), and genevalmag::Attr_grammar::to_string().

Referenced by genevalmag::Maglib::gen_evaluator().

Here is the call graph for this function:



Here is the caller graph for this function:



7.25.4 Member Data Documentation

7.25.4.1 **genevalmag::Parser_AG::attr_grammar** [private]

Internal structure of grammar.

Definition at line 39 of file Parser_AG.h.

Referenced by get_attr_grammar(), parse_grammar(), and save_grammar_file().

7.25.4.2 **genevalmag::Parser_AG::sem_check** [private]

Semantics control for grammar: Well define grammar.

Definition at line 44 of file Parser_AG.h.

Referenced by parse_grammar().

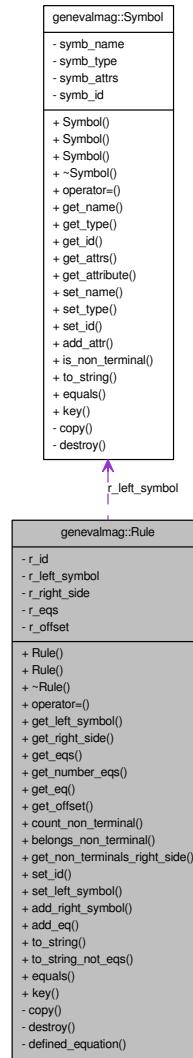
The documentation for this class was generated from the following files:

- [include/Parser/Parser_AG.h](#)
- [src/Parser/Parser_AG.cpp](#)

7.26 genevalmag::Rule Class Reference

```
#include <Rule.h>
```

Collaboration diagram for genevalmag::Rule:



Public Member Functions

- `Rule ()`
- `Rule (const Rule &other)`
- `virtual ~Rule ()`
- `Rule & operator= (const Rule &other)`
- `const Symbol * get_left_symbol () const`
- `const vector< const Symbol * > & get_right_side () const`
- `const map< unsigned short, Equation > & get_eqs () const`
- `const size_t get_number_eqs () const`
- `const Equation * get_eq (const unsigned short index) const`

- const unsigned short `get_offset () const`
- const int `count_non_terminal (const Symbol *symb) const`
- const bool `belongs_non_terminal (const Symbol &non_term) const`
- const vector< const Symbol * > `get_non_terminals_right_side () const`
- void `set_id (const unsigned short id)`
- void `set_left_symbol (const Symbol *left_symb)`
- void `add_right_symbol (const Symbol *right_symb)`
- const bool `add_eq (Equation &eq)`
- const string `to_string () const`
- const string `to_string_not_eqs () const`
- const bool `equals (const Rule &other) const`
- const unsigned short `key () const`

Private Member Functions

- void `copy (const Rule &other)`
- void `destroy ()`
- const bool `defined_equation (const Equation &eq) const`

Private Attributes

- unsigned short `r_id`
Rule's identifier.
- const Symbol * `r_left_symbol`
Rule's left symbol.
- vector< const Symbol * > `r_right_side`
Rule's vector of right symbols.
- map< unsigned short, Equation > `r_eqs`
Rule's map of equations.
- unsigned short `r_offset`
Rule's offset.

7.26.1 Detailed Description

Definition at line 20 of file Rule.h.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 genevalmag::Rule::Rule ()

Constructor empty of rule.

Definition at line 25 of file Rule.cpp.

References r_offset.

7.26.2.2 genevalmag::Rule::Rule (const Rule & other)

Constructor copy of rule.

Parameters

other

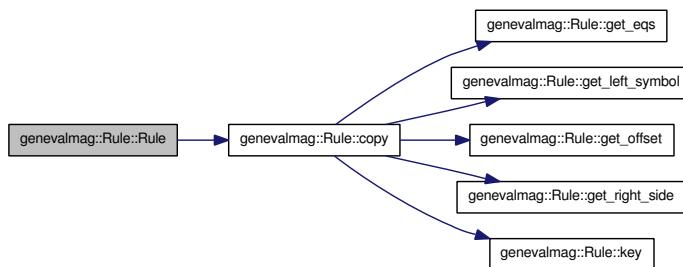
Returns

Constructor copy of rule.

Definition at line 33 of file Rule.cpp.

References copy0.

Here is the call graph for this function:



7.26.2.3 genevalmag::Rule::~Rule () [virtual]

Destructor of rule.

Definition at line 41 of file Rule.cpp.

References destroy0.

Here is the call graph for this function:



7.26.3 Member Function Documentation

7.26.3.1 const bool genevalmag::Rule::add_eq (Equation & eq)

Enqueues a equation in the list of the rule.

Parameters

eq

Returns

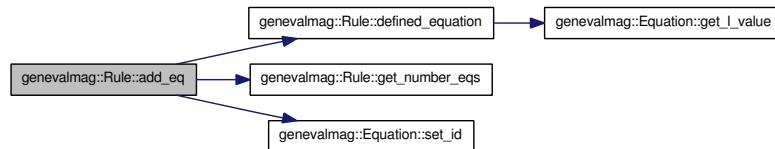
Enqueues a equation in the list of the rule.

Definition at line 195 of file Rule.cpp.

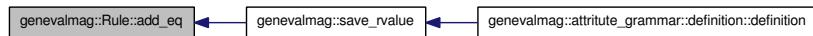
References `defined_equation()`, `get_number_eqs()`, `r_eqs`, `r_offset`, and `genevalmag::Equation::set_id()`.

Referenced by `genevalmag::save_rvalue()`.

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.3.2 void genevalmag::Rule::add_right_symbol (const Symbol * *right_symb*)

Enqueues a symbol in the right side of the rule.

Parameters

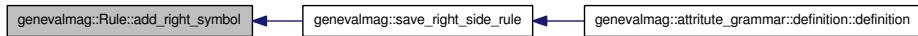
right_symb Enqueues a symbol in the right side of the rule.

Definition at line 153 of file Rule.cpp.

References `r_right_side`.

Referenced by `genevalmag::save_right_side_rule()`.

Here is the caller graph for this function:



7.26.3.3 const bool genevalmag::Rule::belongs_non_terminal (const Symbol & *non_term*) const

Checks that symbol is a non-terminal into the rule.

Parameters

non_term

Returns

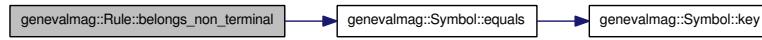
Checks that symbol is a non-terminal into the rule.

Definition at line 176 of file Rule.cpp.

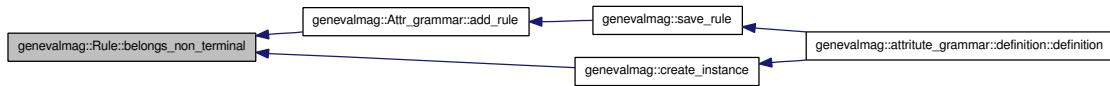
References genevalmag::Symbol::equals(), r_left_symbol, and r_right_side.

Referenced by genevalmag::Attr_grammar::add_rule(), and genevalmag::create_instance().

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.3.4 void genevalmag::Rule::copy (const Rule & other) [private]

Method of copy the rule, STL-like C++.

Parameters

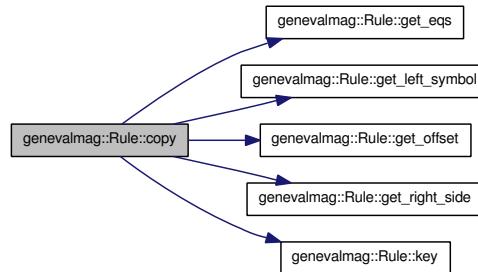
other Method of copy the rule, STL-like C++.

Definition at line 62 of file Rule.cpp.

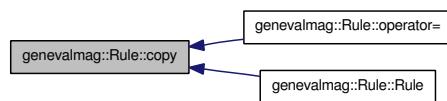
References get_eqs(), get_left_symbol(), get_offset(), get_right_side(), key(), r_eqs, r_id, r_left_symbol, r_offset, and r_right_side.

Referenced by operator=(), and Rule().

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.3.5 **const int genevalmag::Rule::count_non_terminal (const Symbol * *symb*) const**

Returns the count of occurrences of these symbol in the rule.

Parameters

symb

Returns

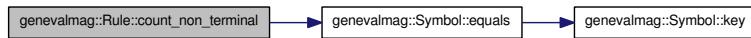
Returns the count of occurrences of these symbol in the rule.

Definition at line 318 of file Rule.cpp.

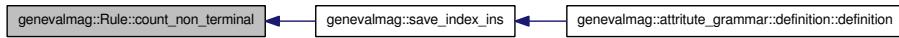
References genevalmag::Symbol::equals(), r_left_symbol, and r_right_side.

Referenced by genevalmag::save_index_ins().

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.3.6 **const bool genevalmag::Rule::defined_equation (const Equation & *eq*) const [private]**

Checks that the equation is not already defined in the rule.

Parameters

eq

Returns

Checks that the equation is not already defined in the rule.

Definition at line 161 of file Rule.cpp.

References genevalmag::Equation::get_l_value(), and r_eqs.

Referenced by add_eq().

Here is the call graph for this function:



Here is the caller graph for this function:



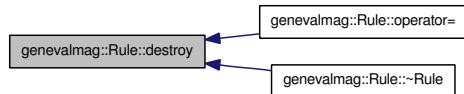
7.26.3.7 void genevalmag::Rule::destroy () [private]

Method destroy rule, STL-like C++.

Definition at line 74 of file Rule.cpp.

Referenced by operator=(), and ~Rule().

Here is the caller graph for this function:



7.26.3.8 const bool genevalmag::Rule::equals (const Rule & other) const

Compares the rule with other.

Parameters

other

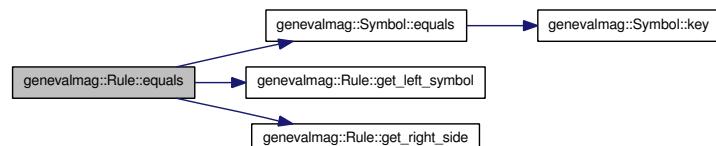
Returns

Compares the rule with other.

Definition at line 281 of file Rule.cpp.

References genevalmag::Symbol::equals(), get_left_symbol(), get_right_side(), r_left_symbol, and r_right_side.

Here is the call graph for this function:



7.26.3.9 const Equation * genevalmag::Rule::get_eq (const unsigned short index) const

Returns the i-equation of the rule.

Parameters***index*****Returns**

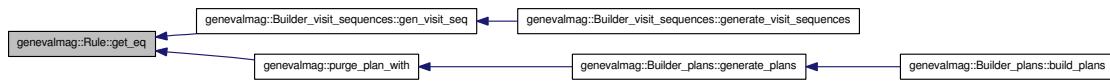
Returns the i-equation of the rule.

Definition at line 121 of file Rule.cpp.

References r_eqs.

Referenced by genevalmag::Builder_visit_sequences::gen_visit_seq(), and genevalmag::purge_plan_with().

Here is the caller graph for this function:



7.26.3.10 const map< unsigned short, Equation > & genevalmag::Rule::get_eqs () const

Returns the equations of the rule.

Returns

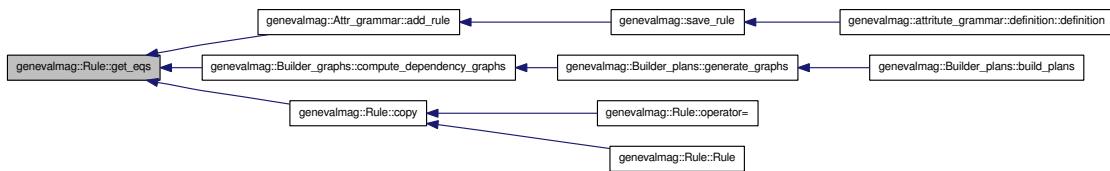
Returns the equations of the rule.

Definition at line 97 of file Rule.cpp.

References r_eqs.

Referenced by genevalmag::Attr_grammar::add_rule(), genevalmag::Builder_graphs::compute_dependency_graphs(), and copy().

Here is the caller graph for this function:



7.26.3.11 const Symbol * genevalmag::Rule::get_left_symbol () const

Returns the left symbol of the rule.

Returns

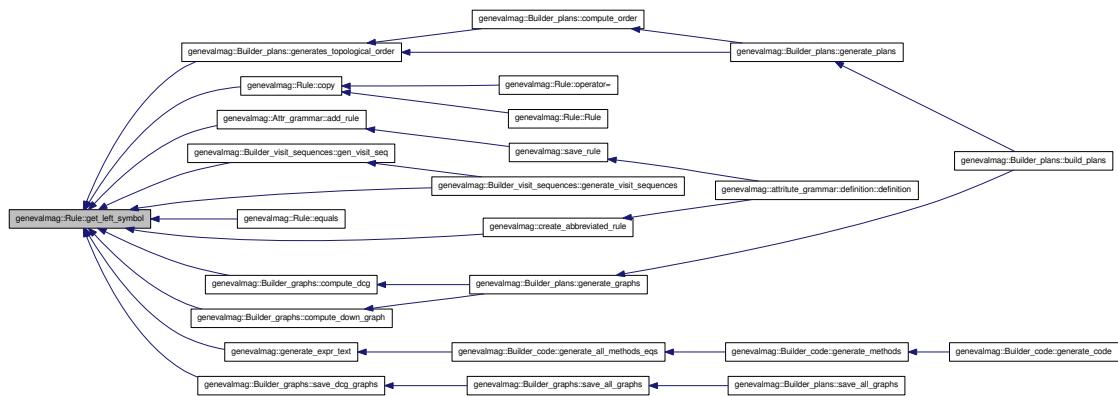
Returns the left symbol of the rule.

Definition at line 81 of file Rule.cpp.

References r_left_symbol.

Referenced by genevalmag::Attr_grammar::add_rule(), genevalmag::Builder_graphs::compute_dcg(), genevalmag::Builder_graphs::compute_down_graph(), copy(), genevalmag::create_abbreviated_rule(), equals(), genevalmag::Builder_visit_sequences::gen_visit_seq(), genevalmag::generate_expr_text(), genevalmag::Builder_visit_sequences::generate_visit_sequences(), genevalmag::Builder_plans::generates_topological_order(), and genevalmag::Builder_graphs::save_dcg_graphs().

Here is the caller graph for this function:



7.26.3.12 const vector< const Symbol * > genevalmag::Rule::get_non_terminals_right_side() const

Returns the non-terminals symbols the right side of the rule.

Returns

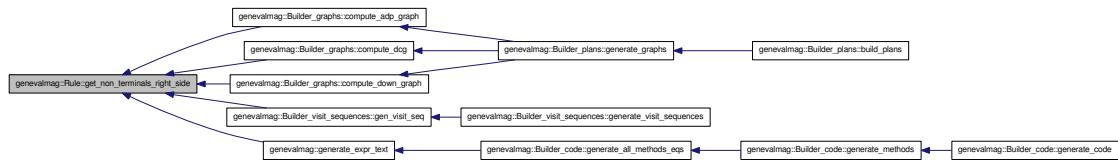
Returns the non-terminals symbols the right side of the rule.

Definition at line 338 of file Rule.cpp.

References r_right_side.

Referenced by genevalmag::Builder_graphs::compute_adp_graph(), genevalmag::Builder_graphs::compute_dcg(), genevalmag::Builder_graphs::compute_down_graph(), genevalmag::Builder_visit_sequences::gen_visit_seq(), and genevalmag::generate_expr_text().

Here is the caller graph for this function:



7.26.3.13 const size_t genevalmag::Rule::get_number_eqs () const

Returns the number of equations of the rule.

Returns

Returns the number of equations of the rule.

Definition at line 105 of file Rule.cpp.

References r_eqs.

Referenced by add_eq0.

Here is the caller graph for this function:



7.26.3.14 const unsigned short genevalmag::Rule::get_offset () const

Returns the offset of all equation of the rule.

Returns

Returns the offset of all equation of the rule.

Definition at line 113 of file Rule.cpp.

References r_offset.

Referenced by copy0.

Here is the caller graph for this function:



7.26.3.15 const vector< const Symbol * > & genevalmag::Rule::get_right_side () const

Returns the right side of the rule.

Returns

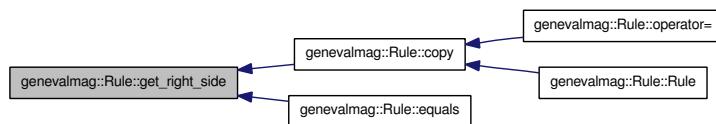
Returns the right side of the rule.

Definition at line 89 of file Rule.cpp.

References r_right_side.

Referenced by copy(), and equals().

Here is the caller graph for this function:

**7.26.3.16 const unsigned short genevalmag::Rule::key () const**

Returns the number key that identifies a rule definitely.

Result = <id_rule>

Returns

Returns the number key that identifies a rule definitely.

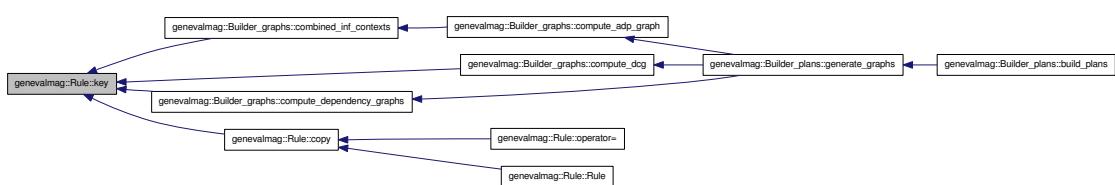
Result = <id_rule>

Definition at line 310 of file Rule.cpp.

References r_id.

Referenced by genevalmag::Builder_graphs::combined_inf_contexts(), genevalmag::Builder_graphs::compute_dcg(), genevalmag::Builder_graphs::compute_dependency_graphs(), and copy().

Here is the caller graph for this function:



7.26.3.17 Rule & genevalmag::Rule::operator= (const Rule & other)

Operator assign(=) of rule.

Parameters

other

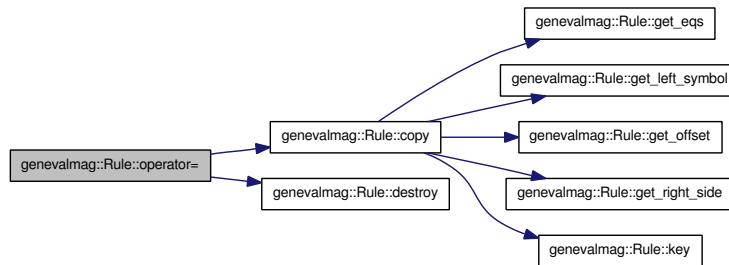
Returns

Operator assign(=) of rule.

Definition at line 49 of file Rule.cpp.

References copy(), and destroy().

Here is the call graph for this function:



7.26.3.18 void genevalmag::Rule::set_id (const unsigned short id)

Sets the identifier of the rule.

Parameters

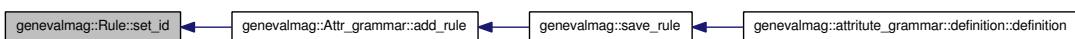
id Sets the identifier of the rule.

Definition at line 137 of file Rule.cpp.

References r_id.

Referenced by genevalmag::Attr_grammar::add_rule().

Here is the caller graph for this function:



7.26.3.19 void genevalmag::Rule::set_left_symbol (const Symbol * left_symb)

Sets the left symbol of the rule.

Parameters

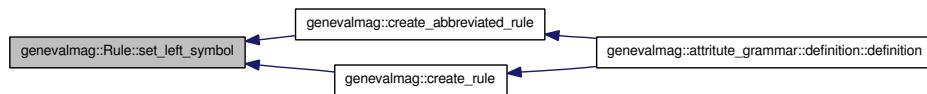
left_symb Sets the left symbol of the rule.

Definition at line 145 of file Rule.cpp.

References r_left_symbol.

Referenced by genevalmag::create_abbreviated_rule(), and genevalmag::create_rule().

Here is the caller graph for this function:

**7.26.3.20 const string genevalmag::Rule::to_string () const**

Generates and returns a string representation of a rule.

Result = left_symbol " ::= " right_side

"compute"

eq_1

...

eq_n

"end;"

where right_ride is = symbol_1 " " ... " " symbol_n

Returns

Generates and returns a string representation of a Rule.

Result = <left_symbol> " ::= " <right_side>

"compute"

eq_1

...

eq_n

"end;"

where <right_ride> is s= symbol_1 " " ... " " symbol_n

Definition at line 229 of file Rule.cpp.

References r_eqs, and to_string_not_eqs().

Here is the call graph for this function:



7.26.3.21 const string genevalmag::Rule::to_string_not_eqs () const

Generates and returns a string representation of a rule.

Result= left_symbol " ::= " right_side ";"

where right_ride is = symbol_1 " " ... " " symbol_n

Returns

Generates and returns a string representation of a rule.

Result = left_symbol " ::= " right_side ";"

where right_ride is= symbol_1 " " ... " " symbol_n

Definition at line 255 of file Rule.cpp.

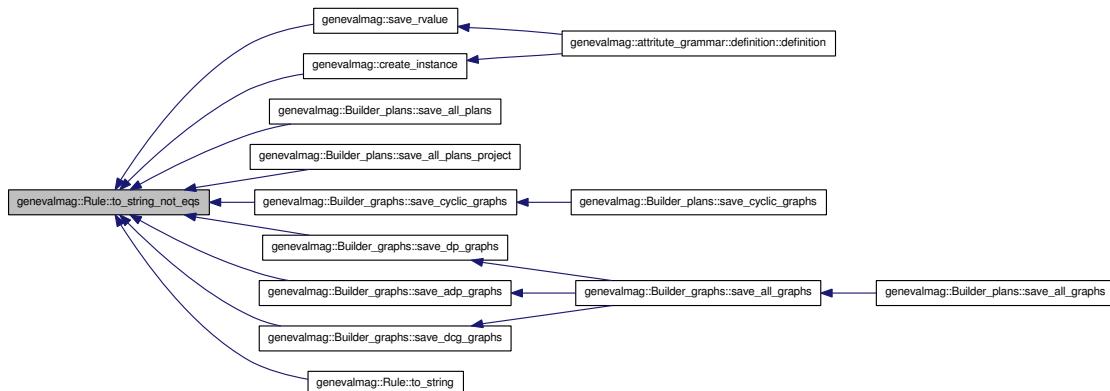
References genevalmag::Symbol::get_name(), r_id, r_left_symbol, and r_right_side.

Referenced by genevalmag::create_instance(), genevalmag::Builder_graphs::save_adp_graphs(), genevalmag::Builder_plans::save_all_plans(), genevalmag::Builder_plans::save_all_plans_project(), genevalmag::Builder_graphs::save_cyclic_graphs(), genevalmag::Builder_graphs::save_dcg_graphs(), genevalmag::Builder_graphs::save_dp_graphs(), genevalmag::save_rvalue(), and to_string().

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.4 Member Data Documentation

7.26.4.1 genevalmag::Rule::r_eqs [private]

Rule's map of equations.

Definition at line 42 of file Rule.h.

Referenced by add_eq(), copy(), defined_equation(), get_eq(), get_eqs(), get_number_eqs(), and to_string().

7.26.4.2 genevalmag::Rule::r_id [private]

Rule's identifier.

Definition at line 27 of file Rule.h.

Referenced by copy(), key(), set_id(), and to_string_not_eqs().

7.26.4.3 genevalmag::Rule::r_left_symbol [private]

Rule's left symbol.

Definition at line 32 of file Rule.h.

Referenced by belongs_non_terminal(), copy(), count_non_terminal(), equals(), get_left_symbol(), set_left_symbol(), and to_string_not_eqs().

7.26.4.4 genevalmag::Rule::r_offset [private]

Rule's offset.

Definition at line 47 of file Rule.h.

Referenced by add_eq(), copy(), get_offset(), and Rule().

7.26.4.5 genevalmag::Rule::r_right_side [private]

Rule's vector of right symbols.

Definition at line 37 of file Rule.h.

Referenced by add_right_symbol(), belongs_non_terminal(), copy(), count_non_terminal(), equals(), get_non_terminals_right_side(), get_right_side(), and to_string_not_eqs().

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

- include/Attr_grammar/Rule.h
- src/Attr_grammar/Rule.cpp

7.27 genevalmag::Semantics_checks Class Reference

```
#include <Semantics_checks.h>
```

Public Member Functions

- `Semantics_checks()`
- `~Semantics_checks()`
- `unsigned short get_precedence_level() const`
- `unsigned short get_index_syntax_order() const`
- `void increment_precedence_level()`
- `void decrement_precedence_level()`
- `void increment_index_syntax_order()`
- `void reset_semantic_context()`
- `void correct_precedence(Expr_function **root_tree)`
- `void correct_associativity(Expr_function **root_tree)`
- `bool check_all_defined_non_terminal(const map<unsigned short, Rule> &rules, const map<string, Symbol> &non_terminals)`
- `bool check_reachability(const map<unsigned short, Rule> &rules, const map<string, Symbol> &non_terminals, const Symbol *init_symbol)`
- `bool check_well_defined_AG(const map<unsigned short, Rule> &rules)`

Private Member Functions

- `void correct_subtree(Expr_function **subtree, int index_root_subtree)`

Private Attributes

- `unsigned short precedence_level`
- `unsigned short index_syntax_order`

Counter of syntax order.

7.27.1 Detailed Description

Definition at line 22 of file Semantics_checks.h.

7.27.2 Constructor & Destructor Documentation

7.27.2.1 genevalmag::Semantics_checks::Semantics_checks()

Constructor empty of Semantics checks.

Constructor empty of Semantic check.

Definition at line 24 of file Semantics_checks.cpp.

References index_syntax_order, and precedence_level.

7.27.2.2 genevalmag::Semantics_checks::~Semantics_checks ()

Destructor of Semantics checks.

Destructor of Semantic check.

Definition at line 33 of file Semantics_checks.cpp.

7.27.3 Member Function Documentation

7.27.3.1 bool genevalmag::Semantics_checks::check_all_defined_non_terminal (const map< unsigned short, Rule > & rules, const map< string, Symbol > & non_terminals)

Verifies that all non-terminals in the grammar has defines in a rule. That is, it is the left value of some rule of grammar.

Parameters

rules

non_terminals

Returns

Verifies that all non-terminals in the grammar has defines in a rule. That is, it is the left value of some rule of grammar.

Definition at line 337 of file Semantics_checks.cpp.

Referenced by genevalmag::check_well_defined().

Here is the caller graph for this function:



7.27.3.2 bool genevalmag::Semantics_checks::check_reachability (const map< unsigned short, Rule > & rules, const map< string, Symbol > & non_terminals, const Symbol * init_symbol)

Computes the boolean adjacency matrix of all the rules of the attributes grammar, then computes the Warshall algorithm for transitive closure and thus can identify from the initial symbol all symbols reachable.

Parameters

rules

non_terminals

init_symbol

Returns

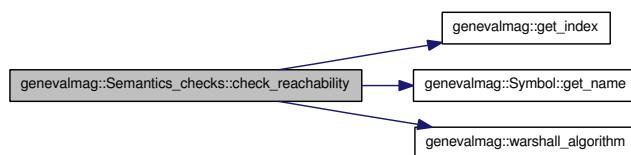
Computes the boolean adjacency matrix of all the rules of the attributes grammar, then computes the Warshall algorithm for transitive closure and thus can identify from the initial symbol all symbols reachable.

Definition at line 406 of file Semantics_checks.cpp.

References genevalmag::get_index(), genevalmag::Symbol::get_name(), and genevalmag::warshall_algorithm().

Referenced by genevalmag::check_well_defined().

Here is the call graph for this function:



Here is the caller graph for this function:



7.27.3.3 bool genevalmag::Semantics_checks::check_well_defined_AG (const map< unsigned short, Rule > & rules)

Checks that all synthesized attributes of the left symbols of all rule are defined with an equation. And all inherit attributes of all right symbol too.

Parameters

rules

Returns

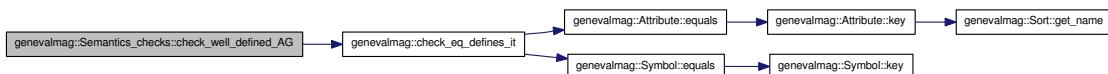
Checks that all synthesized attributes of the left symbols of all rule are defined with an equation. And all inherit attributes of all right symbol too.

Definition at line 487 of file Semantics_checks.cpp.

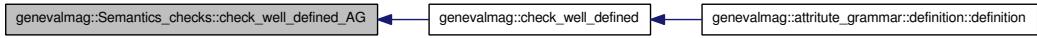
References genevalmag::check_eqDefines_it().

Referenced by genevalmag::check_well_defined().

Here is the call graph for this function:



Here is the caller graph for this function:



7.27.3.4 void genevalmag::Semantics_checks::correct_associativity (Expr_function ** root_tree)

Controls around the tree, that any operator who applies more than once on the same level is associated according to the signature. If it detects conflicts modifying the expression tree with rotations and resources to continue controlling.

Parameters

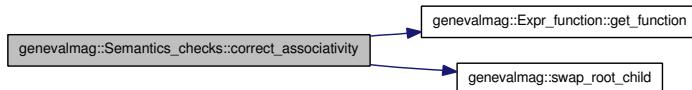
root_tree Controls around the tree, that any operator who applies more than once on the same level is associated according to the signature. If it detects conflicts modifying the expression tree with rotations and resources to continue controlling.

Definition at line 291 of file Semantics_checks.cpp.

References genevalmag::Expr_function::get_function(), genevalmag::k_non_assoc, genevalmag::k_right, and genevalmag::swap_root_child().

Referenced by genevalmag::save_rvalue().

Here is the call graph for this function:



Here is the caller graph for this function:



7.27.3.5 void genevalmag::Semantics_checks::correct_precedence (Expr_function ** root_tree)

Checking from the root of the expression tree to the leaves, which all operators are applies according to their precedence. If there are conflicts resolves them doing rotations, leaving the operator with lower precedence, as the new root.

Obs: The following checks are performed and only make changes in the structure of the tree, if it fulfills the following three conditions:

- Operations that are changed are at the same level of precedence (with respect with parentheses).
- The syntactic order of the expression is not altered.

- The operation with higher precedence apply first.

Parameters

root_tree Checking from the root of the expression tree to the leaves, which all operators are applies according to their precedence. If there are conflicts resolves them doing rotations, leaving the operator with lower precedence, as the new root.

Obs: The following checks are performed and only make changes in the structure of the tree, if it fulfills the following three conditions:

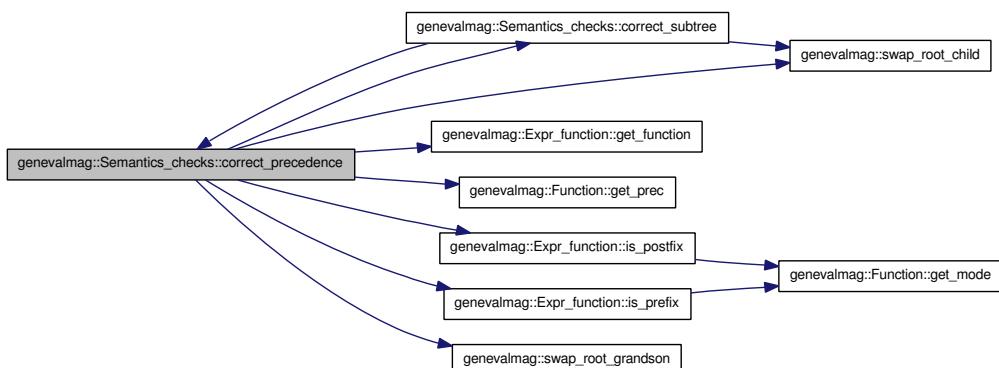
- Operations that are changed are at the same level of precedence (with respect with parentheses).
- The syntactic order of the expression is not altered.
- The operation with higher precedence apply first.

Definition at line 209 of file Semantics_checks.cpp.

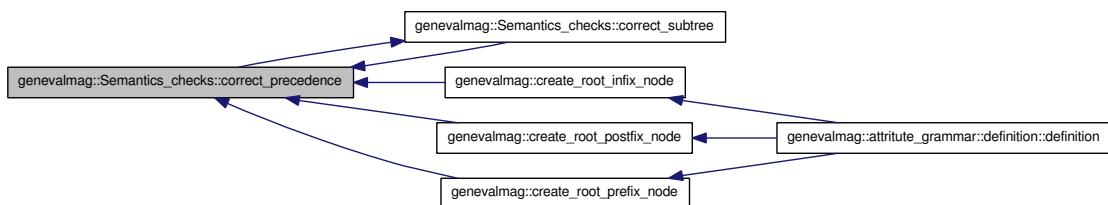
References correct_subtree(), genevalmag::Expr_function::get_function(),
genevalmag::Function::get_prec(), genevalmag::Expr_function::is_postfix(),
genevalmag::Expr_function::is_prefix(), genevalmag::swap_root_child(), and
genevalmag::swap_root_grandson().

Referenced by correct_subtree(), genevalmag::create_root_infix_node(), genevalmag::create_root_postfix_node(), and genevalmag::create_root_prefix_node().

Here is the call graph for this function:



Here is the caller graph for this function:



7.27.3.6 void genevalmag::Semantics_checks::correct_subtree (Expr_function ** subtree, int index_root_subtree) [private]

Check and correct the precedence of the operator in a subtree.

Parameters

subtree

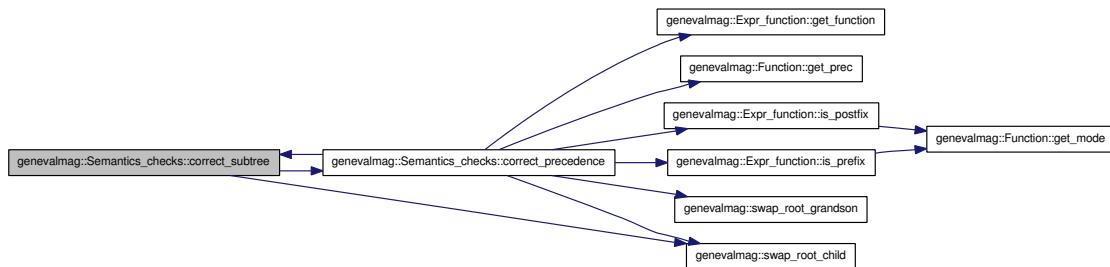
index_root_subtree Checks and correct the precedence of the operator in a subtree.

Definition at line 172 of file Semantics_checks.cpp.

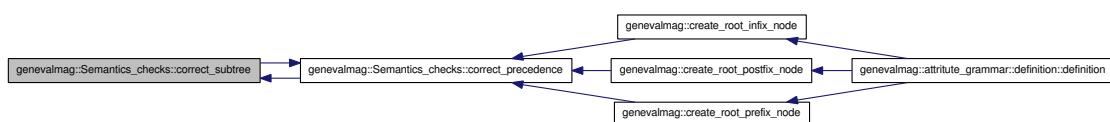
References correct_precedence(), and genevalmag::swap_root_child().

Referenced by correct_precedence().

Here is the call graph for this function:



Here is the caller graph for this function:



7.27.3.7 void genevalmag::Semantics_checks::decrement_precedence_level ()

Decrements the level because a parenthesis closing.

Definition at line 64 of file Semantics_checks.cpp.

References precedence_level.

Referenced by genevalmag::decrement_level().

Here is the caller graph for this function:



7.27.3.8 `unsigned short genevalmag::Semantics_checks::get_index_syntax_order () const`

Returns the current index of the syntax order.

Returns

Returns the current index of the syntax order.

Definition at line 48 of file Semantics_checks.cpp.

References index_syntax_order.

Referenced by genevalmag::create_func_node().

Here is the caller graph for this function:



7.27.3.9 `unsigned short genevalmag::Semantics_checks::get_precedence_level () const`

Returns the precedence level.

Returns

Returns the precedence level.

Definition at line 40 of file Semantics_checks.cpp.

References precedence_level.

Referenced by genevalmag::create_func_node().

Here is the caller graph for this function:



7.27.3.10 `void genevalmag::Semantics_checks::increment_index_syntax_order ()`

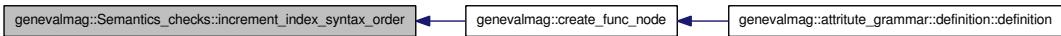
Increments the syntax order global.

Definition at line 72 of file Semantics_checks.cpp.

References index_syntax_order.

Referenced by genevalmag::create_func_node().

Here is the caller graph for this function:



7.27.3.11 void genevalmag::Semantics_checks::increment_precedence_level ()

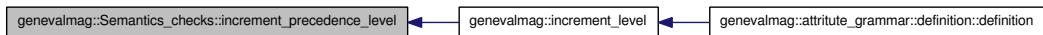
Increments the level because a new parenthesis opening.

Definition at line 56 of file Semantics_checks.cpp.

References precedence_level.

Referenced by genevalmag::increment_level().

Here is the caller graph for this function:



7.27.3.12 void genevalmag::Semantics_checks::reset_semantic_context ()

Resets all variables that affect in the precedence analysys.

Definition at line 80 of file Semantics_checks.cpp.

References index_syntax_order, and precedence_level.

Referenced by genevalmag::save_rvalue().

Here is the caller graph for this function:



7.27.4 Member Data Documentation

7.27.4.1 genevalmag::Semantics_checks::index_syntax_order [private]

Counter of syntax order.

Definition at line 35 of file Semantics_checks.h.

Referenced by get_index_syntax_order(), increment_index_syntax_order(), reset_semantic_context(), and Semantics_checks().

7.27.4.2 unsigned short genevalmag::Semantics_checks::precedence_level [private]

Definition at line 29 of file Semantics_checks.h.

Referenced by decrement_precedence_level(), get_precedence_level(), increment_precedence_level(), reset_semantic_context(), and Semantics_checks().

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

- include/Parser/Semantics_checks.h
- src/Parser/Semantics_checks.cpp

7.28 genevalmag::skip_parser Struct Reference

Classes

- struct definition

7.28.1 Detailed Description

Declaration of a parser for inputs that ignore within an [Attribute](#) Grammar file, with the Spirit library of Boost.

Definition at line 66 of file Parser_AG.cpp.

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

- src/Parser/[Parser_AG.cpp](#)

7.29 `genevalmag::skip_parser::definition< ScannerT >` Struct Template Reference

Public Member Functions

- `definition (skip_parser const &self)`
- `rule< ScannerT > const & start () const`

Public Attributes

- `rule< ScannerT > skip`

7.29.1 Detailed Description

```
template<typename ScannerT> struct genevalmag::skip_parser::definition< ScannerT >
```

Definition at line 69 of file Parser_AG.cpp.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 `template<typename ScannerT > genevalmag::skip_parser::definition< ScannerT >::definition (skip_parser const & self) [inline]`

Definition at line 71 of file Parser_AG.cpp.

References `genevalmag::skip_parser::definition< ScannerT >::skip`.

7.29.3 Member Function Documentation

7.29.3.1 `template<typename ScannerT > rule<ScannerT> const& genevalmag::skip_parser::definition< ScannerT >::start () const [inline]`

Definition at line 81 of file Parser_AG.cpp.

References `genevalmag::skip_parser::definition< ScannerT >::skip`.

7.29.4 Member Data Documentation

7.29.4.1 `template<typename ScannerT > rule<ScannerT> genevalmag::skip_parser::definition< ScannerT >::skip`

Definition at line 79 of file Parser_AG.cpp.

Referenced by `genevalmag::skip_parser::definition< ScannerT >::definition()`, and `genevalmag::skip_parser::definition< ScannerT >::start()`.

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

- src/Parser/Parser_AG.cpp

7.30 genevalmag::Sort Class Reference

```
#include <Sort.h>
```

Public Member Functions

- `Sort ()`
- `Sort (const string name)`
- `Sort (const Sort &other)`
- `virtual ~Sort ()`
- `Sort & operator= (const Sort &other)`
- `const string get_name () const`
- `void set_name (const string name)`
- `void set_type_basic (const bool type_basic)`
- `const string to_string () const`
- `const bool equals (const Sort &other) const`
- `const bool is_type_basic () const`
- `const string key () const`

Private Member Functions

- `void copy (const Sort &other)`
- `void destroy ()`

Private Attributes

- `string s_name`
Sort's name.
- `unsigned int s_ins`
Sort's instance.
- `bool s_type_basic`
Is true iff is a type basic of C++.

7.30.1 Detailed Description

Definition at line 19 of file Sort.h.

7.30.2 Constructor & Destructor Documentation

7.30.2.1 genevalmag::Sort::Sort ()

Constructor empty of `Sort`.

Definition at line 20 of file Sort.cpp.

References `s_ins`.

7.30.2.2 genevalmag::Sort::Sort (*const string name*)

Constructor with the name of the Sort.

Parameters

name

Returns

Constructor with the name of the Sort.

Definition at line 28 of file Sort.cpp.

References s_ins, s_name, and s_type_basic.

7.30.2.3 genevalmag::Sort::Sort (*const Sort & other*)

Constructor copy of Sort.

Parameters

other

Returns

Constructor copy of Sort.

Definition at line 38 of file Sort.cpp.

References copy().

Here is the call graph for this function:



7.30.2.4 genevalmag::Sort::~Sort () [virtual]

Destructor of Sort.

Definition at line 46 of file Sort.cpp.

References destroy().

Here is the call graph for this function:



7.30.3 Member Function Documentation

7.30.3.1 void genevalmag::Sort::copy (const Sort & other) [private]

Method of copy the [Sort](#), STL-like C++.

Parameters

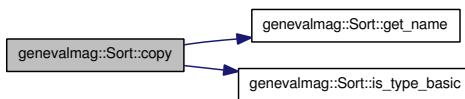
other Method of copy the [Sort](#), STL-like C++.

Definition at line 67 of file Sort.cpp.

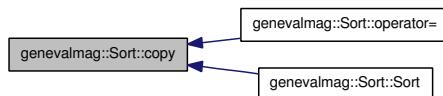
References `get_name()`, `is_type_basic()`, `s_ins`, `s_name`, and `s_type_basic`.

Referenced by `operator=()`, and `Sort()`.

Here is the call graph for this function:



Here is the caller graph for this function:



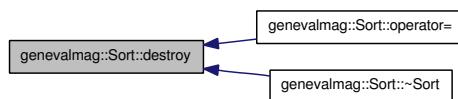
7.30.3.2 void genevalmag::Sort::destroy () [private]

Method destroy [Sort](#), STL-like C++.

Definition at line 79 of file Sort.cpp.

Referenced by `operator=()`, and `~Sort()`.

Here is the caller graph for this function:



7.30.3.3 const bool genevalmag::Sort::equals (const Sort & other) const

Compares the [Sort](#) with other.

Parameters

other

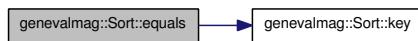
Returns

Compares the [Sort](#) with other.

Definition at line 141 of file Sort.cpp.

References key().

Here is the call graph for this function:

**7.30.3.4 const string genevalmag::Sort::get_name () const**

Returns the name of the [Sort](#).

Returns

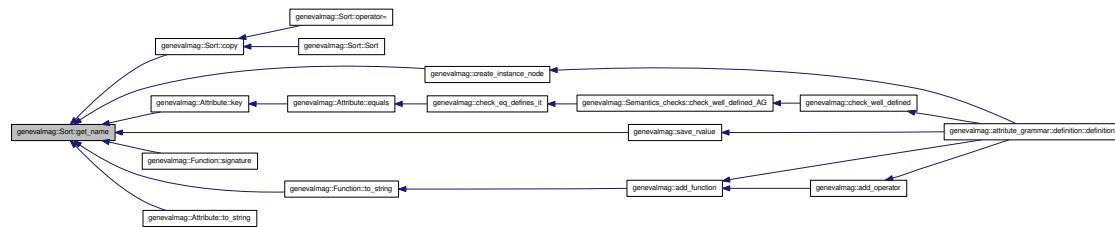
Returns the name of the [Sort](#).

Definition at line 86 of file Sort.cpp.

References s_name.

Referenced by copy(), genevalmag::create_instance_node(), genevalmag::Attribute::key(), genevalmag::save_rvalue(), genevalmag::Function::signature(), genevalmag::Function::to_string(), and genevalmag::Attribute::to_string().

Here is the caller graph for this function:

**7.30.3.5 const bool genevalmag::Sort::is_type_basic () const**

Returns if the [Sort](#) is basic.

Returns

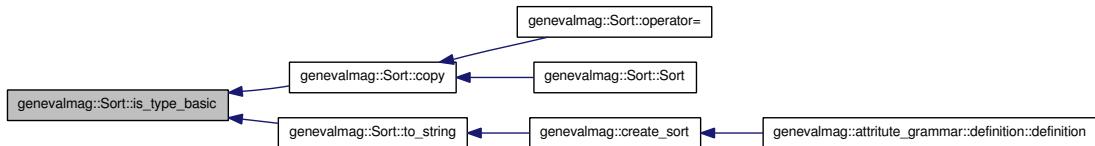
Returns if the [Sort](#) is basic.

Definition at line 149 of file Sort.cpp.

References `s_type_basic`.

Referenced by `copy()`, and `to_string()`.

Here is the caller graph for this function:



7.30.3.6 const string genevalmag::Sort::key () const

Generates and returns the string key that identifies a `Sort` definitely.

Result = <name>

Returns

Generates and returns the string key that identifies a `Sort` definitely.

Result = <name>

Definition at line 159 of file `Sort.cpp`.

References `s_name`.

Referenced by `equals()`.

Here is the caller graph for this function:



7.30.3.7 Sort & genevalmag::Sort::operator= (const Sort & other)

Operator assign(=) of `Sort`.

Parameters

other

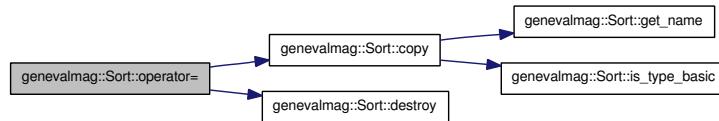
Returns

Operator assign(=) of `Sort`.

Definition at line 54 of file `Sort.cpp`.

References `copy()`, and `destroy()`.

Here is the call graph for this function:



7.30.3.8 void genevalmag::Sort::set_name (const string *name*)

Sets the name of the Sort.

Parameters

name Sets the name of the Sort.

Definition at line 94 of file Sort.cpp.

References s_name.

7.30.3.9 void genevalmag::Sort::set_type_basic (const bool *type_basic*)

Sets if the Sort is basic.

Parameters

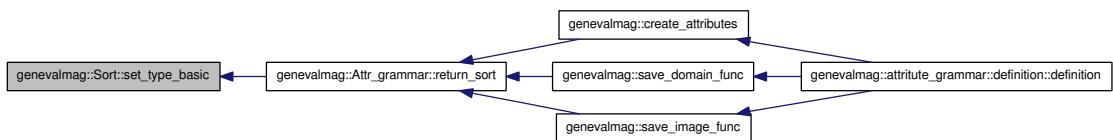
type_basic Set if the Sort is basic.

Definition at line 101 of file Sort.cpp.

References s_type_basic.

Referenced by genevalmag::Attr_grammar::return_sort().

Here is the caller graph for this function:



7.30.3.10 const string genevalmag::Sort::to_string () const

Generates and returns a string representation of a Sort.

Result = "sort" <name> ["(" <instance> ")" IF DEBUG IS ON] ";"

Returns

Generates and returns a string representation of a [Sort](#).

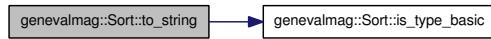
Result = "sort" <name> ["(" <instance> ")" IF DEBUG IS ON] ":"

Definition at line 111 of file Sort.cpp.

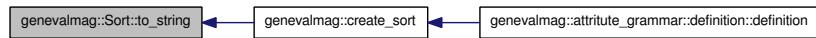
References [is_type_basic\(\)](#), [s_ins](#), and [s_name](#).

Referenced by [genevalmag::create_sort\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



7.30.4 Member Data Documentation

7.30.4.1 [genevalmag::Sort::s_ins](#) [private]

Sort's instance.

Definition at line 31 of file Sort.h.

Referenced by [copy\(\)](#), [Sort\(\)](#), and [to_string\(\)](#).

7.30.4.2 [genevalmag::Sort::s_name](#) [private]

Sort's name.

Definition at line 26 of file Sort.h.

Referenced by [copy\(\)](#), [get_name\(\)](#), [key\(\)](#), [set_name\(\)](#), [Sort\(\)](#), and [to_string\(\)](#).

7.30.4.3 [genevalmag::Sort::s_type_basic](#) [private]

Is true iff is a type basic of C++.

Definition at line 36 of file Sort.h.

Referenced by [copy\(\)](#), [is_type_basic\(\)](#), [set_type_basic\(\)](#), and [Sort\(\)](#).

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

- [include/Attr_grammar/Sort.h](#)
- [src/Attr_grammar/Sort.cpp](#)

7.31 genevalmag::Symbol Class Reference

```
#include <Symbol.h>
```

Public Member Functions

- `Symbol ()`
- `Symbol (const string &name, const symbol_type &type)`
- `Symbol (const Symbol &other)`
- `virtual ~Symbol ()`
- `Symbol & operator= (const Symbol &other)`
- `string get_name () const`
- `symbol_type get_type () const`
- `unsigned short get_id () const`
- `const vector< const Attribute * > & getAttrs () const`
- `const Attribute * get_attribute (const string &name_attr) const`
- `void set_name (const string name)`
- `void set_type (const symbol_type type)`
- `void set_id (unsigned short id)`
- `void add_attr (const Attribute *attr)`
- `const bool is_non_terminal () const`
- `string to_string () const`
- `bool equals (const Symbol &other) const`
- `string key () const`

Private Member Functions

- `void copy (const Symbol &other)`
- `void destroy ()`

Private Attributes

- `string symb_name`
Symbol's name.
- `symbol_type symb_type`
Symbol's type.
- `vector< const Attribute * > symbAttrs`
Symbol's vector attribute.
- `unsigned short symb_id`
Symbol's identifier.

7.31.1 Detailed Description

Definition at line 29 of file Symbol.h.

7.31.2 Constructor & Destructor Documentation

7.31.2.1 genevalmag::Symbol::Symbol ()

Constructor empty of [Symbol](#).

Definition at line 19 of file Symbol.cpp.

7.31.2.2 genevalmag::Symbol::Symbol (const string & name, const symbol_type & type)

Constructor with name and type of [Symbol](#).

Parameters

name

type

Returns

Constructor with name and type of [Symbol](#).

Definition at line 26 of file Symbol.cpp.

References symb_id, symb_name, and symb_type.

7.31.2.3 genevalmag::Symbol::Symbol (const Symbol & other)

Constructor copy of [Symbol](#).

Parameters

other

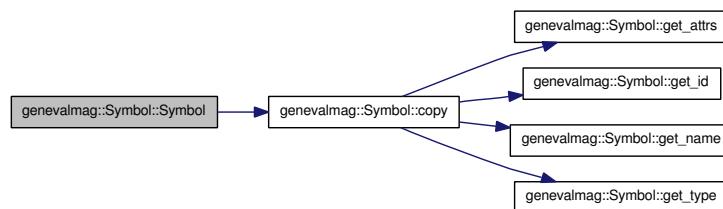
Returns

Constructor copy of [Symbol](#).

Definition at line 36 of file Symbol.cpp.

References copy().

Here is the call graph for this function:



7.31.2.4 genevalmag::Symbol::~Symbol () [virtual]

Destructor of [Symbol](#).

Definition at line 44 of file [Symbol.cpp](#).

References [destroy\(\)](#).

Here is the call graph for this function:



7.31.3 Member Function Documentation

7.31.3.1 void genevalmag::Symbol::add_attr (const Attribute * attr)

Enqueues a attribute in the list of the [Symbol](#).

Parameters

attr Enqueues a attribute in the list of the symbol.

Definition at line 155 of file [Symbol.cpp](#).

References [symbAttrs](#).

Referenced by [genevalmag::Attr_grammar::load_attributes\(\)](#).

Here is the caller graph for this function:



7.31.3.2 void genevalmag::Symbol::copy (const Symbol & other) [private]

Method of copy the symbol, STL-like C++.

Parameters

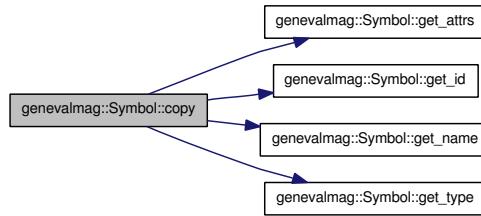
other Method of copy the [Symbol](#), STL-like C++.

Definition at line 65 of file [Symbol.cpp](#).

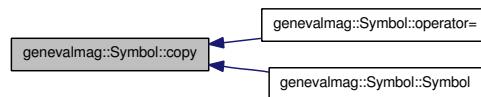
References [getAttrs\(\)](#), [getId\(\)](#), [getName\(\)](#), [getType\(\)](#), [symbAttrs](#), [symbId](#), [symbName](#), and [symbType](#).

Referenced by [operator=\(\)](#), and [Symbol\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



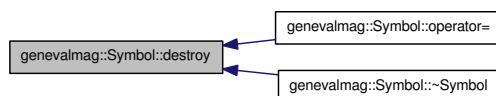
7.31.3.3 void genevalmag::Symbol::destroy () [private]

Method destroy symbol, STL-like C++.

Definition at line 76 of file Symbol.cpp.

Referenced by operator=(), and ~Symbol().

Here is the caller graph for this function:



7.31.3.4 bool genevalmag::Symbol::equals (const Symbol & other) const

Compares the [Symbol](#) with other.

Parameters

other

Returns

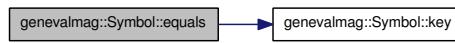
Compares the symbol with other.

Definition at line 207 of file Symbol.cpp.

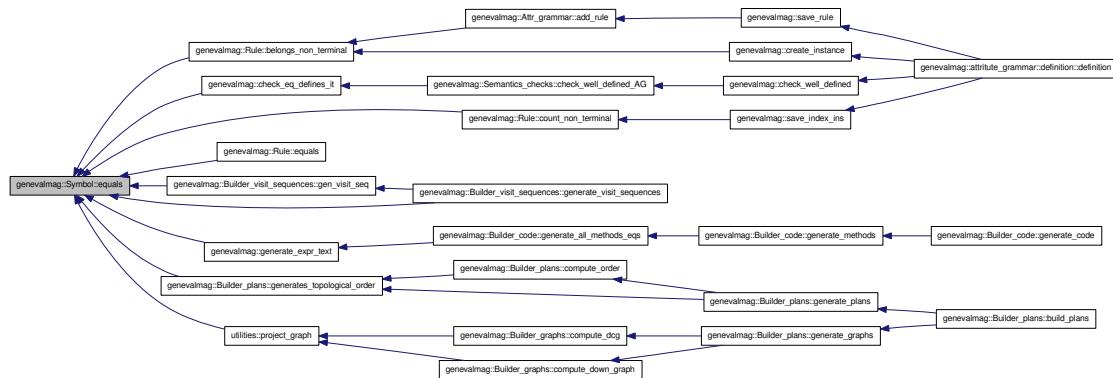
References key().

Referenced by genevalmag::Rule::belongs_non_terminal(), genevalmag::check_eqDefines_it(), genevalmag::Rule::count_non_terminal(), genevalmag::Rule::equals(), genevalmag::Builder_visit_sequences::gen_visit_seq(), genevalmag::generate_expr_text(), genevalmag::Builder_visit_sequences::generate_visit_sequences(), genevalmag::Builder_plans::generates_topological_order(), and utilities::project_graph().

Here is the call graph for this function:



Here is the caller graph for this function:



7.31.3.5 const Attribute * genevalmag::Symbol::get_attribute (const string & name_attr) const

Finds in the list of attribute of the symbol and returns the attribute with that name passed as parameter.

Parameters

name_attr

Returns

Finds in the list of attribute of the symbol and returns the attribute with that name passed as parameter.

Definition at line 116 of file Symbol.cpp.

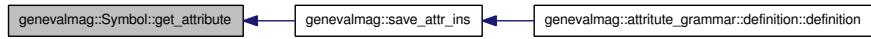
References get_name(), and symbAttrs.

Referenced by genevalmag::saveAttrIns().

Here is the call graph for this function:



Here is the caller graph for this function:



7.31.3.6 `const vector< const Attribute * > & genevalmag::Symbol::getAttrs() const`

Returns the list of attributes of the [Symbol](#).

Returns

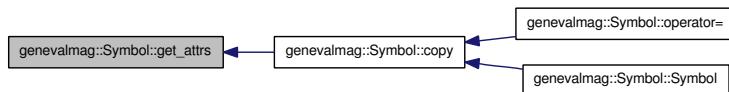
Returns the list of attributes of the symbol.

Definition at line 107 of file [Symbol.cpp](#).

References [symbAttrs](#).

Referenced by [copy\(\)](#).

Here is the caller graph for this function:



7.31.3.7 `unsigned short genevalmag::Symbol::getId() const`

Returns the id of the [Symbol](#).

Returns

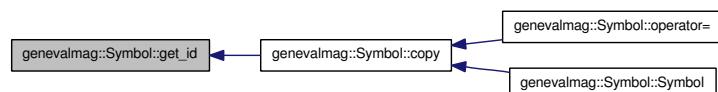
Returns the id of the symbol.

Definition at line 91 of file [Symbol.cpp](#).

References [symbId](#).

Referenced by [copy\(\)](#).

Here is the caller graph for this function:



7.31.3.8 string genevalmag::Symbol::get_name () const

Returns the name of the [Symbol](#).

Returns

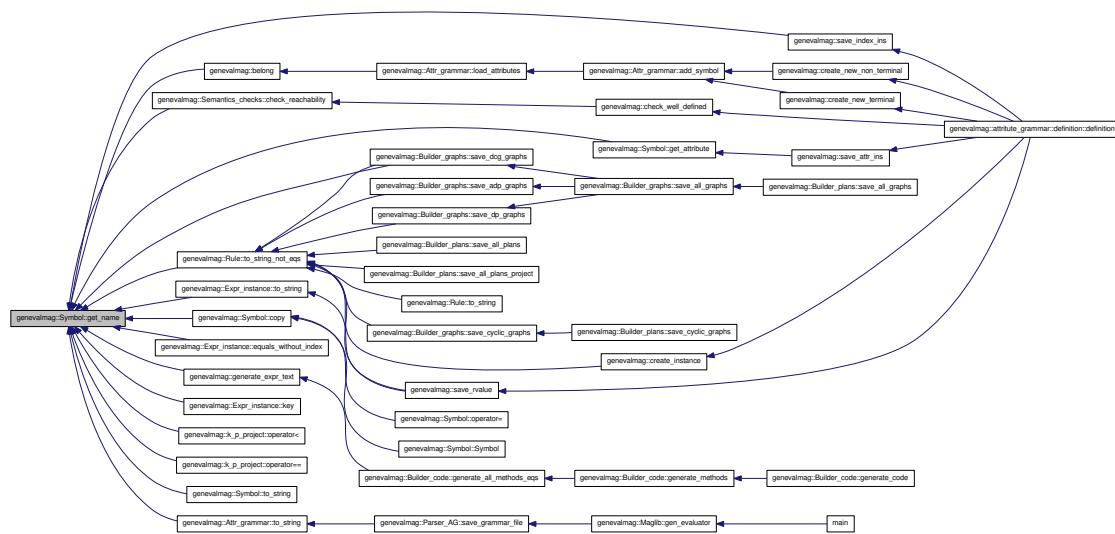
Returns the name of the symbol.

Definition at line 83 of file [Symbol.cpp](#).

References [symb_name](#).

Referenced by [genevalmag::belong\(\)](#), [genevalmag::Semantics_checks::check_reachability\(\)](#), [copy\(\)](#), [genevalmag::Expr_instance::equals_without_index\(\)](#), [genevalmag::generate_expr_text\(\)](#), [get_attribute\(\)](#), [genevalmag::Expr_instance::key\(\)](#), [genevalmag::k_p_project::operator<\(\)](#), [genevalmag::k_p_project::operator==\(\)](#), [genevalmag::Builder_graphs::save_dcg_graphs\(\)](#), [genevalmag::save_index_ins\(\)](#), [genevalmag::Expr_instance::to_string\(\)](#), [to_string\(\)](#), [genevalmag::Attr_grammar::to_string\(\)](#), and [genevalmag::Rule::to_string_not_eqs\(\)](#).

Here is the caller graph for this function:



7.31.3.9 symbol_type genevalmag::Symbol::get_type () const

Returns the type of the [Symbol](#).

Returns

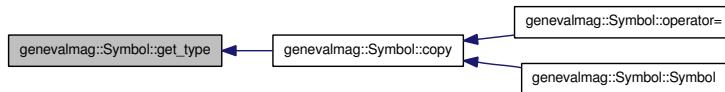
Returns the type of the symbol.

Definition at line 99 of file [Symbol.cpp](#).

References [symb_type](#).

Referenced by copy0.

Here is the caller graph for this function:



7.31.3.10 const bool genevalmag::Symbol::is_non_terminal () const

Returns true if the Symbol's type is Non Terminal.

Returns

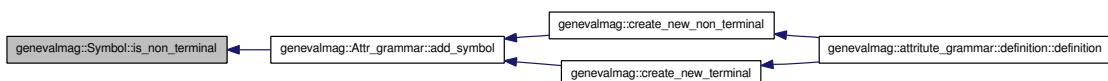
Returns true if the symbol's type is non-terminal.

Definition at line 163 of file Symbol.cpp.

References genevalmag::k_non_terminal, and symb_type.

Referenced by genevalmag::Attr_grammar::add_symbol().

Here is the caller graph for this function:



7.31.3.11 string genevalmag::Symbol::key () const

Generates and returns the string key that identifies a [Symbol](#) definitely.

Result= <name>

Returns

Generates and returns the string key that identifies a symbol definitely.

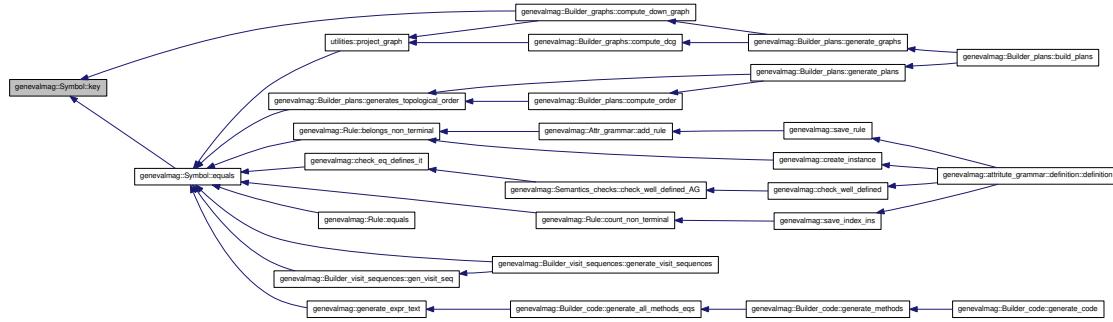
Result = name

Definition at line 217 of file Symbol.cpp.

References symb_name.

Referenced by genevalmag::Builder_graphs::compute_down_graph(), and equals().

Here is the caller graph for this function:



7.31.3.12 Symbol & genevalmag::Symbol::operator= (const Symbol & other)

Operator assign(=) of [Symbol](#).

Parameters

other

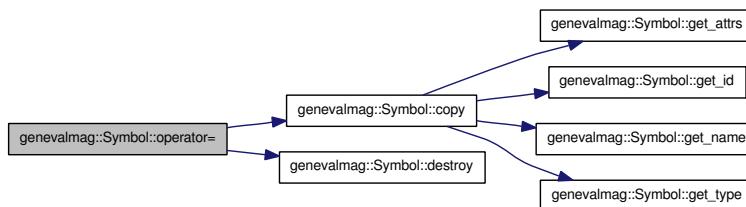
Returns

Operator assign(=) of [Symbol](#).

Definition at line 52 of file [Symbol.cpp](#).

References [copy\(\)](#), and [destroy\(\)](#).

Here is the call graph for this function:



7.31.3.13 void genevalmag::Symbol::set_id (unsigned short id)

Sets the id of the [Symbol](#).

Parameters

id Sets the id of the symbol.

Definition at line 147 of file [Symbol.cpp](#).

References [symb_id](#).

7.31.3.14 void genevalmag::Symbol::set_name (const string name)

Sets the name of the [Symbol](#).

Parameters

name Sets the name of the symbol.

Definition at line 131 of file Symbol.cpp.

References symb_name.

7.31.3.15 void genevalmag::Symbol::set_type (const symbol_type type)

Sets the type of the [Symbol](#).

Parameters

type Sets the type of the symbol.

Definition at line 139 of file Symbol.cpp.

References symb_type.

7.31.3.16 string genevalmag::Symbol::to_string () const

Generates and returns a string representation of a [Symbol](#).

Result= "symbol" <name> <type> ";"

where if <type> is Non Terminal, then list the attributes of the symbol.

list = "Attributes:" attr_1 "," ... "," attr_n

Returns

Generates and returns a string representation of a symbol.

Result = "symbol" name type ";"

where if 'type' is Non Terminal, then list the attributes of the symbol.

list = "Attributes:" attr_1 "," ... "," attr_n

Definition at line 177 of file Symbol.cpp.

References get_name(), genevalmag::k_non_terminal, genevalmag::k_terminal, symbAttrs, symb_name, and symb_type.

Here is the call graph for this function:



7.31.4 Member Data Documentation

7.31.4.1 genevalmag::Symbol::symb_attrs [private]

Symbol's vector attribute.

Definition at line 46 of file Symbol.h.

Referenced by add_attr(), copy(), get_attribute(), getAttrs(), and to_string().

7.31.4.2 genevalmag::Symbol::symb_id [private]

Symbol's identifier.

Definition at line 51 of file Symbol.h.

Referenced by copy(), get_id(), setId(), and Symbol().

7.31.4.3 genevalmag::Symbol::symb_name [private]

Symbol's name.

Definition at line 36 of file Symbol.h.

Referenced by copy(), getName(), key(), setName(), Symbol(), and to_string().

7.31.4.4 genevalmag::Symbol::symb_type [private]

Symbol's type.

Definition at line 41 of file Symbol.h.

Referenced by copy(), getType(), isNonTerminal(), setType(), Symbol(), and to_string().

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

- include/Attr_grammar/Symbol.h
- src/Attr_grammar/Symbol.cpp

7.32 genevalmag::vertex_data_t Struct Reference

```
#include <Builder_graphs.h>
```

Public Types

- `typedef vertex_property_tag kind`

7.32.1 Detailed Description

Definition at line 28 of file `Builder_graphs.h`.

7.32.2 Member Typedef Documentation

7.32.2.1 `typedef vertex_property_tag genevalmag::vertex_data_t::kind`

Definition at line 30 of file `Builder_graphs.h`.

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

- `include/Builders/Builder_graphs.h`

Chapter 8

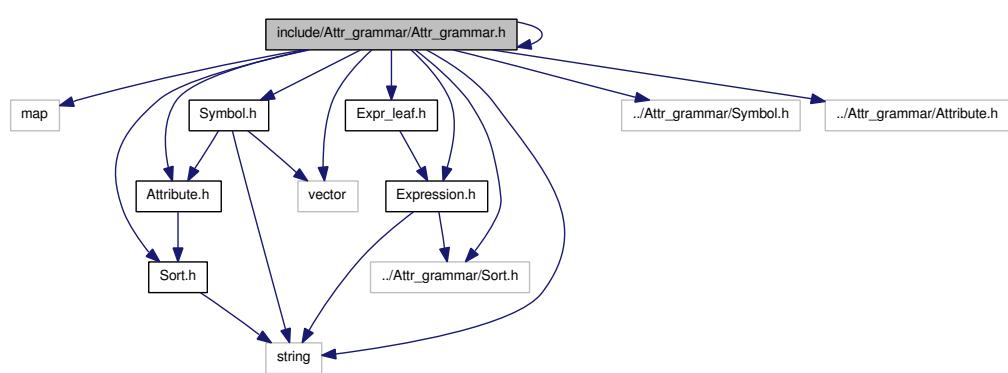
File Documentation

8.1 include/Attr_grammar/Attr_grammar.h File Reference

Class that represent the full attribute grammar.

```
#include <map>
#include "Sort.h"
#include <string>
#include <vector>
#include "Sort.h"
#include "Attribute.h"
#include "Symbol.h"
#include "../Attr_grammar/Sort.h"
#include "Expression.h"
#include "Expr_leaf.h"
#include "../Attr_grammar/Symbol.h"
#include "../Attr_grammar/Attribute.h"
```

Include dependency graph for Attr_grammar.h:



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



Classes

- class [genevalmag::Attr_grammar](#)

Namespaces

- namespace [genevalmag](#)

8.1.1 Detailed Description

Class that represent the full attribute grammar.

Date

28/09/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

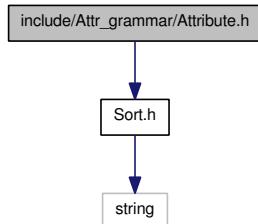
Definition in file [Attr_grammar.h](#).

8.2 include/Attr_grammar/Attribute.h File Reference

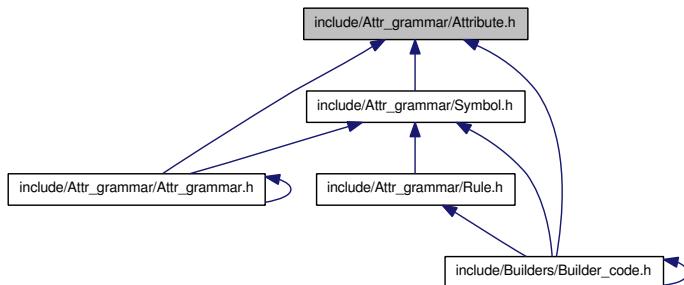
Class of the attribute of the attribute grammar.

```
#include "Sort.h"
```

Include dependency graph for Attribute.h:



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



Classes

- class `genevalmag::Attribute`

Namespaces

- namespace `genevalmag`

Enumerations

- enum `genevalmag::type_attr` { `genevalmag::k_inherit`, `genevalmag::k_synthetize` }

8.2.1 Detailed Description

Class of the attribute of the attribute grammar.

Date

23/10/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

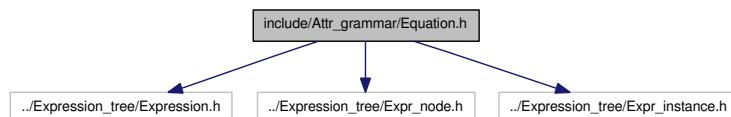
Definition in file [Attribute.h](#).

8.3 include/Attr_grammar/Equation.h File Reference

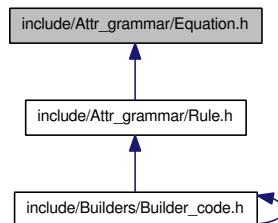
Class of the equations of the attribute grammar.

```
#include "../Expression_tree/Expression.h"
#include "../Expression_tree/Expr_node.h"
#include "../Expression_tree/Expr_instance.h"
```

Include dependency graph for Equation.h:



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



Classes

- class `genevalmag::Equation`

Namespaces

- namespace `genevalmag`

8.3.1 Detailed Description

Class of the equations of the attribute grammar.

Date

05/11/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

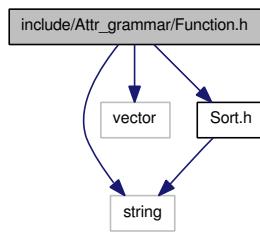
Definition in file [Equation.h](#).

8.4 include/Attr_grammar/Function.h File Reference

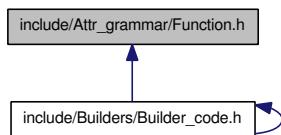
Class of the function of the attribute grammar.

```
#include <string>
#include <vector>
#include "Sort.h"
```

Include dependency graph for Function.h:



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



Classes

- class `genevalmag::Function`

Namespaces

- namespace `genevalmag`

Enumerations

- enum `genevalmag::oper_mode` { `genevalmag::k_prefix`, `genevalmag::k_infix`, `genevalmag::k_postfix` }
- enum `genevalmag::oper_assoc` { `genevalmag::k_left`, `genevalmag::k_right`, `genevalmag::k_non_assoc` }

8.4.1 Detailed Description

Class of the function of the attribute grammar.

Date

26/11/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

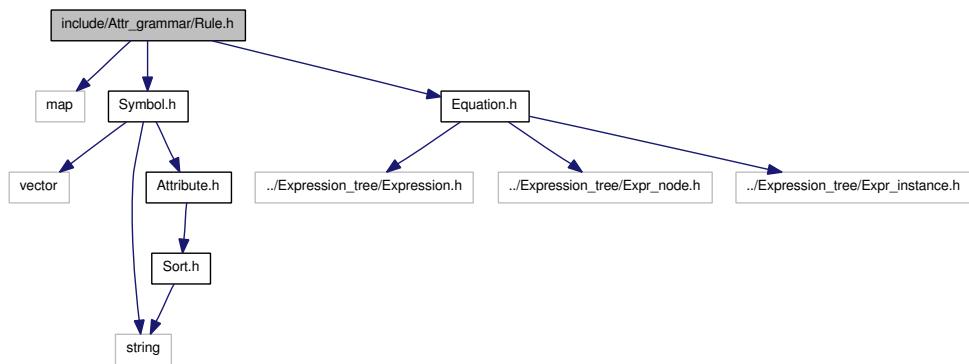
Definition in file [Function.h](#).

8.5 include/Attr_grammar/Rule.h File Reference

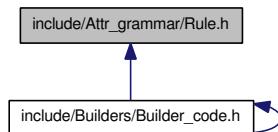
Class of the rule of the attribute grammar.

```
#include <map>
#include "Symbol.h"
#include "Equation.h"
```

Include dependency graph for Rule.h:



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



Classes

- class genevalmag::Rule

Namespaces

- namespace genevalmag

8.5.1 Detailed Description

Class of the rule of the attribute grammar.

Date

05/11/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

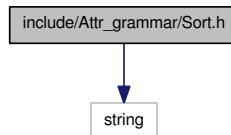
Definition in file [Rule.h](#).

8.6 include/Attr_grammar/Sort.h File Reference

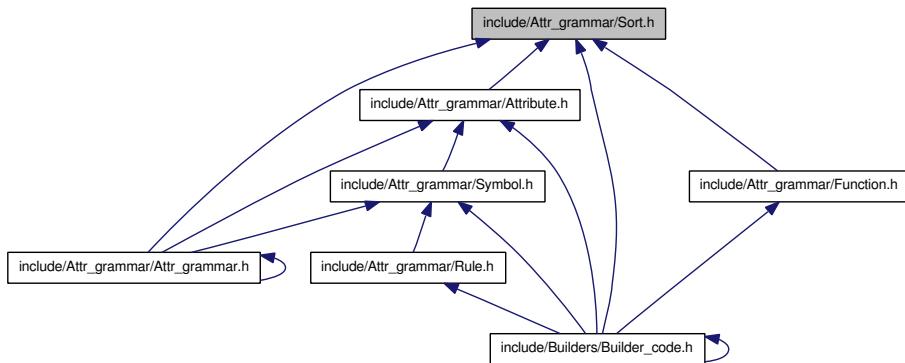
Class of the sort of the attribute grammar.

```
#include <string>
```

Include dependency graph for Sort.h:



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



Classes

- class `genevalmag::Sort`

Namespaces

- namespace `genevalmag`

8.6.1 Detailed Description

Class of the sort of the attribute grammar.

Date

28/09/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

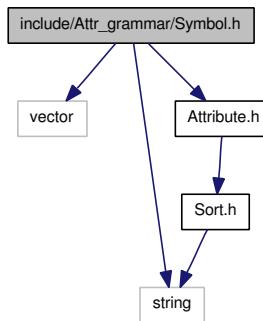
Definition in file [Sort.h](#).

8.7 include/Attr_grammar/Symbol.h File Reference

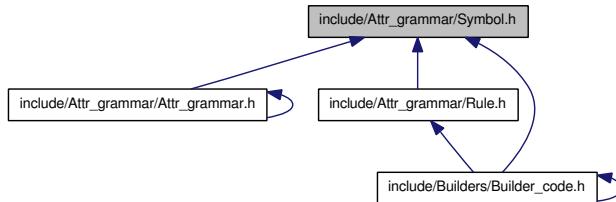
Class of the symbol of the attribute grammar.

```
#include <vector>
#include <string>
#include "Attribute.h"
```

Include dependency graph for Symbol.h:



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



Classes

- class `genevalmag::Symbol`

Namespaces

- namespace `genevalmag`

Enumerations

- enum `genevalmag::symbol_type` { `genevalmag::k_terminal`, `genevalmag::k_non_terminal` }

8.7.1 Detailed Description

Class of the symbol of the attribute grammar.

Date

04/11/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

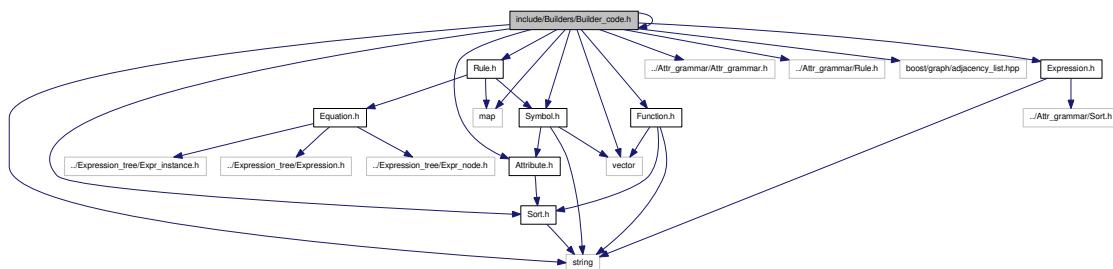
Definition in file [Symbol.h](#).

8.8 include/Builders/Builder_code.h File Reference

Header with the functions for build the header and source code of the evaluator of MAG.

```
#include "../Attr_grammar/Attr_grammar.h"
#include <map>
#include "Sort.h"
#include "Function.h"
#include "Attribute.h"
#include "Symbol.h"
#include "Rule.h"
#include "../Attr_grammar/Attr_grammar.h"
#include "../Attr_grammar/Rule.h"
#include <string>
#include <vector>
#include <boost/graph/adjacency_list.hpp>
#include "Expression.h"
```

Include dependency graph for Builder_code.h:



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



Classes

- class genevalmag::Builder_code

Namespaces

- namespace genevalmag

8.8.1 Detailed Description

Header with the functions for build the header and source code of the evaluator of MAG.

Date

18/03/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

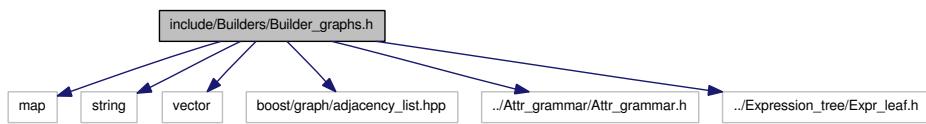
Definition in file [Builder_code.h](#).

8.9 include/Builders/Builder_graphs.h File Reference

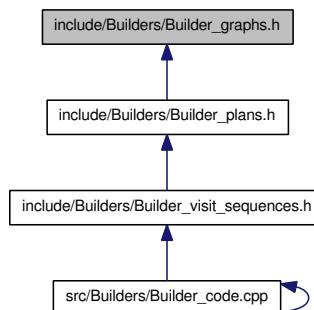
Header with the functions for build all kind of graphs for the attribute grammar.

```
#include <map>
#include <string>
#include <vector>
#include <boost/graph/adjacency_list.hpp>
#include "../Attr_grammar/Attr_grammar.h"
#include "../Expression_tree/Expr_leaf.h"
```

Include dependency graph for Builder_graphs.h:



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



Classes

- struct genevalmag::vertex_data_t
- class genevalmag::Builder_graphs

Namespaces

- namespace genevalmag

Typedefs

- typedef adjacency_list< hash_setS, vecS, directedS, property_vertex_dp > genevalmag::Graph
- typedef Graph::vertex_descriptor genevalmag::Vertex

Variables

- `typedef< vertex_data_t, constgenevalmag::Expr_leaf * > genevalmag::property_vertex_dp`

8.9.1 Detailed Description

Header with the functions for build all kind of graphs for the attribute grammar.

Date

17/02/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

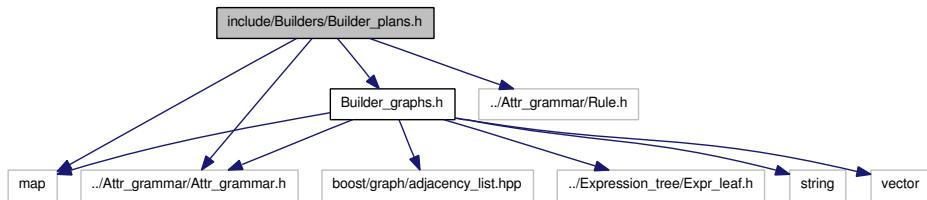
Definition in file [Builder_graphs.h](#).

8.10 include/Builders/Builder_plans.h File Reference

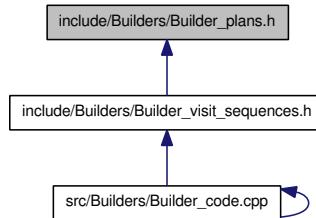
Header of generation plans module of Attribute Grammar.

```
#include <map>
#include "../Attr_grammar/Attr_grammar.h"
#include "../Attr_grammar/Rule.h"
#include "Builder_graphs.h"
```

Include dependency graph for Builder_plans.h:



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



Classes

- struct `genevalmag::c_rule`
- struct `genevalmag::k_w`
- struct `genevalmag::i_w`
- struct `genevalmag::k_plan`
- struct `genevalmag::k_p_project`
- class `genevalmag::Builder_plans`

Namespaces

- namespace `genevalmag`

Typedefs

- typedef `vector< unsigned short >` `genevalmag::Order_eval_eq`
- typedef `vector< unsigned short >` `genevalmag::Order_rule`
- typedef struct `genevalmag::c_rule` `genevalmag::Context_rule`

- typedef struct genevalmag::k_w genevalmag::Key_work_list
- typedef struct genevalmag::i_w genevalmag::Item_work
- typedef struct genevalmag::k_plan genevalmag::Key_plan
- typedef struct genevalmag::k_p_project genevalmag::Key_plan_project

8.10.1 Detailed Description

Header of generation plans module of Attribute Grammar.

Date

18/02/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

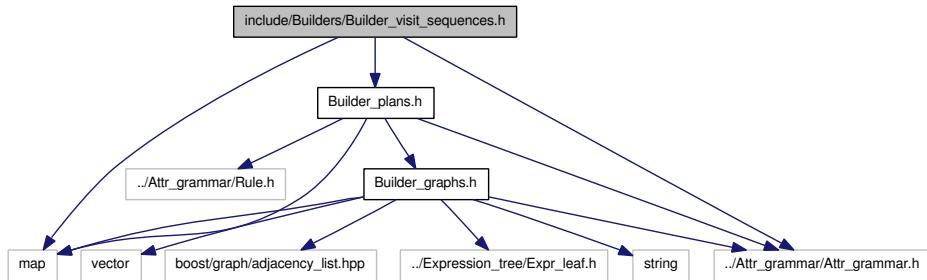
Definition in file [Builder_plans.h](#).

8.11 include/Builders/Builder_visit_sequences.h File Reference

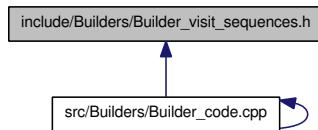
Header with the functions for build the visits sequences of grammar.

```
#include <map>
#include "../Attr_grammar/Attr_grammar.h"
#include "Builder_plans.h"
```

Include dependency graph for Builder_visit_sequences.h:



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



Classes

- class `genevalmag::Builder_visit_sequences`

Namespaces

- namespace `genevalmag`

Typedefs

- typedef `vector< int > genevalmag::Visit_seq`

8.11.1 Detailed Description

Header with the functions for build the visits sequences of grammar. Implementation of the methods the `Builder_visit_sequences.h`.

Date

17/03/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

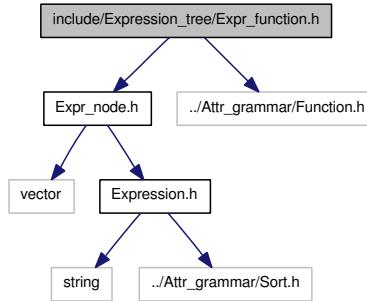
Definition in file [Builder_visit_sequences.h](#).

8.12 include/Expression_tree/Expr_function.h File Reference

Function element of an Expression.

```
#include "Expr_node.h"
#include "../Attr_grammar/Function.h"
```

Include dependency graph for Expr_function.h:



Classes

- class genevalmag::Expr_function

Namespaces

- namespace genevalmag

Functions

- const bool genevalmag::IS_OPERATOR (true)

8.12.1 Detailed Description

Function element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

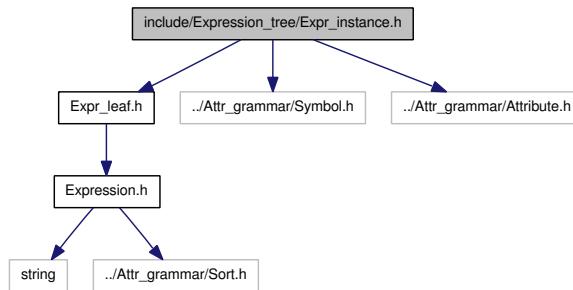
Definition in file [Expr_function.h](#).

8.13 include/Expression_tree/Expr_instance.h File Reference

Instance element of an Expression.

```
#include "Expr_leaf.h"
#include "../Attr_grammar/Symbol.h"
#include "../Attr_grammar/Attribute.h"
```

Include dependency graph for Expr_instance.h:



Classes

- class genevalmag::Expr_instance

Namespaces

- namespace genevalmag

8.13.1 Detailed Description

Instance element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

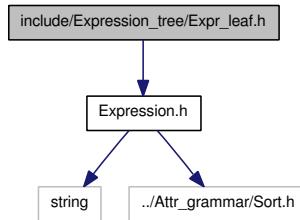
Definition in file [Expr_instance.h](#).

8.14 include/Expression_tree/Expr_leaf.h File Reference

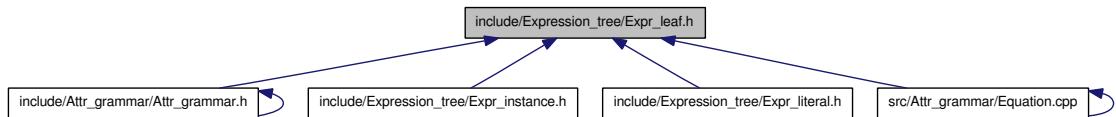
Abstract final element of an Expression.

```
#include "Expression.h"
```

Include dependency graph for Expr_leaf.h:



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



Classes

- class `genevalmag::Expr_leaf`

Namespaces

- namespace `genevalmag`

8.14.1 Detailed Description

Abstract final element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

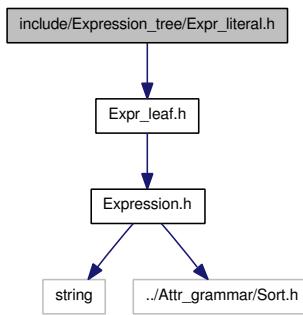
Definition in file `Expr_leaf.h`.

8.15 include/Expression_tree/Expr_literal.h File Reference

Literal element of an Expression.

```
#include "Expr_leaf.h"
```

Include dependency graph for Expr_literal.h:



Classes

- class genevalmag::Expr_literal

Namespaces

- namespace genevalmag

Enumerations

- enum genevalmag::literal_type {
 genevalmag::k_int, genevalmag::k_float, genevalmag::k_char, genevalmag::k_string,
 genevalmag::k_bool }

8.15.1 Detailed Description

Literal element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

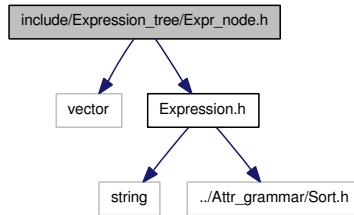
Definition in file [Expr_literal.h](#).

8.16 include/Expression_tree/Expr_node.h File Reference

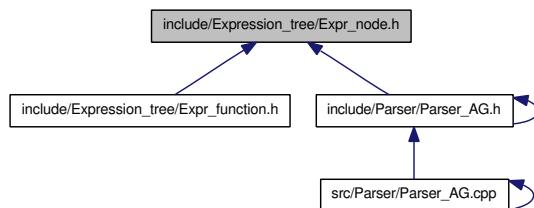
Abstract recursive element of an Expression.

```
#include <vector>
#include "Expression.h"
```

Include dependency graph for Expr_node.h:



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



Classes

- class genevalmag::Expr_node

Namespaces

- namespace genevalmag

8.16.1 Detailed Description

Abstract recursive element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

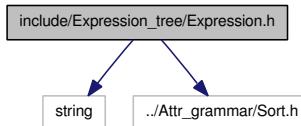
Definition in file [Expr_node.h](#).

8.17 include/Expression_tree/Expression.h File Reference

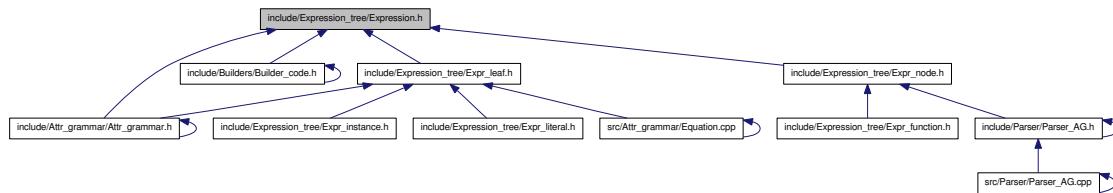
Abstract expression.

```
#include <string>
#include "../Attr_grammar/Sort.h"
```

Include dependency graph for Expression.h:



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



Classes

- class genevalmag::Expression

Namespaces

- namespace genevalmag

8.17.1 Detailed Description

Abstract expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

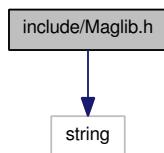
Definition in file [Expression.h](#).

8.18 include/Maglib.h File Reference

Header of the library of evaluator's generator.

```
#include <string>
```

Include dependency graph for Maglib.h:



Classes

- class [genevalmag::Maglib](#)

Namespaces

- namespace [genevalmag](#)

8.18.1 Detailed Description

Header of the library of evaluator's generator.

Date

14/04/2010

Author

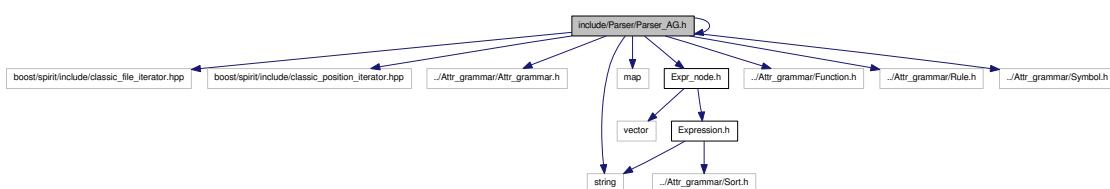
Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

Definition in file [Maglib.h](#).

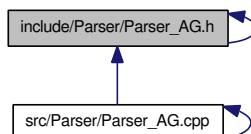
8.19 include/Parser/Parser_AG.h File Reference

```
#include <boost/spirit/include/classic_file_iterator.hpp>
#include <boost/spirit/include/classic_position_iterator.hpp>
#include "../Attr_grammar/Attr_grammar.h"
#include "Semantics_checks.h"
#include <string>
#include <map>
#include "Expr_node.h"
#include "../Attr_grammar/Function.h"
#include "../Attr_grammar/Rule.h"
#include "../Attr_grammar/Symbol.h"
```

Include dependency graph for Parser_AG.h:



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



Classes

- class genevalmag::Parser_AG

Namespaces

- namespace genevalmag

Typedefs

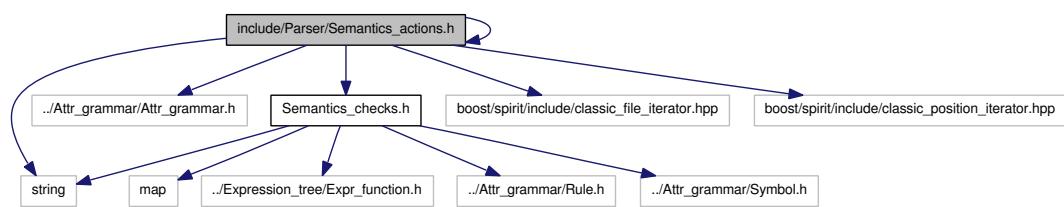
- typedef char genevalmag::char_t
- typedef file_iterator<char_t> genevalmag::iterator_f
- typedef position_iterator<iterator_f> genevalmag::iterator_t

8.20 include/Parser/Semantics_actions.h File Reference

Header semantics actions for parse of Attribute grammar.

```
#include <string>
#include "../Attr_grammar/Attr_grammar.h"
#include "Semantics_checks.h"
#include "Parser_AG.h"
#include <boost/spirit/include/classic_file_iterator.hpp>
#include <boost/spirit/include/classic_position_iterator.hpp>
```

Include dependency graph for Semantics_actions.h:



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



Namespaces

- namespace genevalmag

Functions

- void genevalmag::set_attr_grammar (Attr_grammar *at_grammar)
- void genevalmag::set_sem_check (Semantics_checks *s_check)
- void genevalmag::create_sort (const iterator_t str, const iterator_t end)
- void genevalmag::inic_func (const iterator_t str, const iterator_t end)
- void genevalmag::add_function (const iterator_t str, const iterator_t end)
- void genevalmag::save_name_func (const iterator_t str, const iterator_t end)
- void genevalmag::save_domain_func (const iterator_t str, const iterator_t end)
- void genevalmag::save_image_func (const iterator_t str, const iterator_t end)
- void genevalmag::add_operator (const iterator_t str, const iterator_t end)
- void genevalmag::save_mode_op (const iterator_t str, const iterator_t end)
- void genevalmag::save_prec_op (int const prec)
- void genevalmag::save_assoc_op (const iterator_t str, const iterator_t end)
- void genevalmag::add_attribute (const iterator_t str, const iterator_t end)
- void genevalmag::save_sort_attr (const iterator_t str, const iterator_t end)
- void genevalmag::save_type_attr (const iterator_t str, const iterator_t end)

- void `genevalmag::save_member_list_attr` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_attributes` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_new_non_terminal` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_new_terminal` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_rule` (const iterator_t str, const iterator_t end)
- void `genevalmag::save_right_side_rule` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_abbreviated_rule` (const iterator_t str, const iterator_t end)
- void `genevalmag::save_rule` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_instance` (const iterator_t str, const iterator_t end)
- void `genevalmag::save_index_ins` (int const index)
- void `genevalmag::save_attr_ins` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_lit_number` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_lit_ch` (const iterator_t ch, const iterator_t end)
- void `genevalmag::create_lit_str` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_bool` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_function` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_operator` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_equation` (const iterator_t str, const iterator_t end)
- void `genevalmag::save_rvalue` (const iterator_t str, const iterator_t end)
- void `genevalmag::push_mark` (char name)
- void `genevalmag::create_literal_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_instance_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_func_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_root_infix_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_root_function_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_root_postfix_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_root_prefix_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::check_well_defined` (const iterator_t str, const iterator_t end)
- void `genevalmag::increment_level` (char name)
- void `genevalmag::decrement_level` (char name)

8.20.1 Detailed Description

Header semantics actions for parse of Attribute grammar.

Date

09/02/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

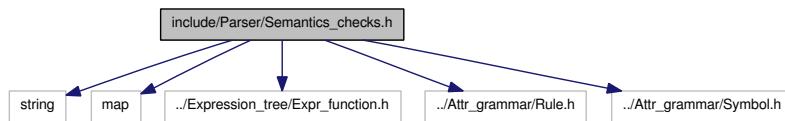
Definition in file [Semantics_actions.h](#).

8.21 include/Parser/Semantics_checks.h File Reference

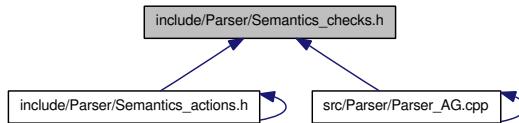
Header method Semantics checks of Attribute grammar.

```
#include <string>
#include <map>
#include "../Expression_tree/Expr_function.h"
#include "../Attr_grammar/Rule.h"
#include "../Attr_grammar/Symbol.h"
```

Include dependency graph for Semantics_checks.h:



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



Classes

- class [genevalmag::Semantics_checks](#)

Namespaces

- namespace [genevalmag](#)

8.21.1 Detailed Description

Header method Semantics checks of Attribute grammar.

Date

11/02/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

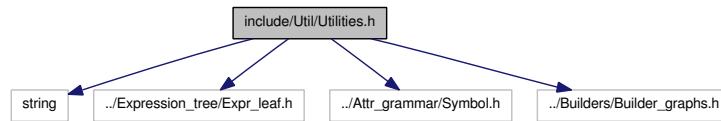
Definition in file [Semantics_checks.h](#).

8.22 include/Util/Utilities.h File Reference

Header of utilities module, where are methods and function used by many class.

```
#include <string>
#include "../Expression_tree/Expr_leaf.h"
#include "../Attr_grammar/Symbol.h"
#include "../Builders/Builder_graphs.h"
```

Include dependency graph for Utilities.h:



Namespaces

- namespace `utilities`

Functions

- bool `utilities::create_folder (const string path)`
- bool `utilities::clean_output_folder (const string path)`
- bool `utilities::copy_static_code (const string path_d, const string path_s)`
- void `utilities::generate_names_instance (const Graph &graph, string datas[], size_t size_d)`
- void `utilities::generate_names_attr (const Graph &graph, string datas[], size_t size_d)`
- void `utilities::print_graph (const Graph &graph, const string path, const string name_file, const string name_graph, const string names[], string shape_vertex)`
- Vertex `utilities::return_vertex (const Graph &graph, const Expr_leaf *node)`
- void `utilities::merge_graph (const Graph &graph1, const Graph &graph2, Graph &graph_merged)`
- void `utilities::project_graph (const Symbol *symb, Graph &graph)`
- string `utilities::cleaning_tabs (const string str)`
- string `utilities::write_inf_context (const vector< unsigned short > &context_vec)`
- bool `utilities::belong_index (const unsigned short &index, const vector< unsigned short > &vec)`

8.22.1 Detailed Description

Header of utilities module, where are methods and function used by many class. Implementation of utilities module, where are methods and function used by many class.

Date

09/03/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

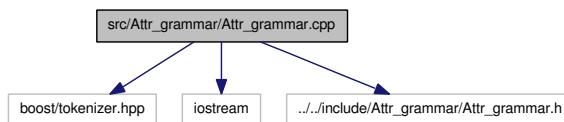
Definition in file [Utilities.h](#).

8.23 src/Attr_grammar/Attr_grammar.cpp File Reference

Implementation of the methods the [Attr_grammar.h](#).

```
#include <boost/tokenizer.hpp>
#include <iostream>
#include "../../include/Attr_grammar/Attr_grammar.h"
```

Include dependency graph for Attr_grammar.cpp:



Namespaces

- namespace [genevalmag](#)

Functions

- template<class K , class T >
const bool [genevalmag::add](#) (const T &elem, map< K, T > &map_elem)
- template<class K , class T >
const string [genevalmag::to_string_map](#) (const map< K, T > &map_elem)
- const bool [genevalmag::belong](#) (const Symbol &symb, const string &exprAttrs)

8.23.1 Detailed Description

Implementation of the methods the [Attr_grammar.h](#).

Date

28/09/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

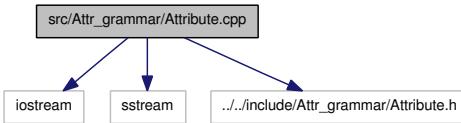
Definition in file [Attr_grammar.cpp](#).

8.24 src/Attr_grammar/Attribute.cpp File Reference

Implementation of the methods the [Attribute.h](#).

```
#include <iostream>
#include <sstream>
#include "../../include/Attr_grammar/Attribute.h"
```

Include dependency graph for Attribute.cpp:



Namespaces

- namespace [genevalmag](#)

8.24.1 Detailed Description

Implementation of the methods the [Attribute.h](#).

Date

23/10/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

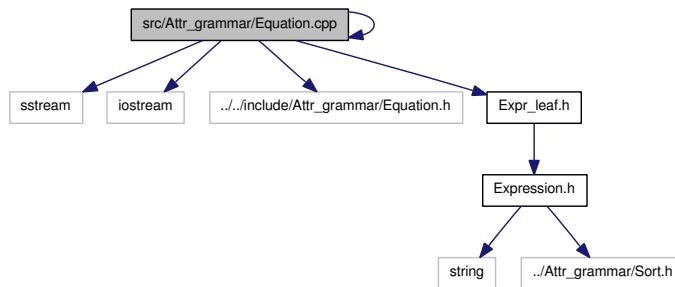
Definition in file [Attribute.cpp](#).

8.25 src/Attr_grammar/Equation.cpp File Reference

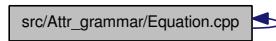
Implementation of the methods the [Equation.h](#).

```
#include <sstream>
#include <iostream>
#include "../../include/Attr_grammar/Equation.h"
#include "../../include/Expression_tree/Expr_literal.h"
#include "Expr_leaf.h"
```

Include dependency graph for Equation.cpp:



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



Namespaces

- namespace [genevalmag](#)

8.25.1 Detailed Description

Implementation of the methods the [Equation.h](#).

Date

05/11/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

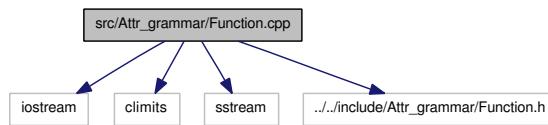
Definition in file [Equation.cpp](#).

8.26 src/Attr_grammar/Function.cpp File Reference

Implementation of the methods the [Function.h](#).

```
#include <iostream>
#include <climits>
#include <sstream>
#include "../include/Attr_grammar/Function.h"
```

Include dependency graph for Function.cpp:



Namespaces

- namespace [genevalmag](#)

8.26.1 Detailed Description

Implementation of the methods the [Function.h](#).

Date

26/11/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

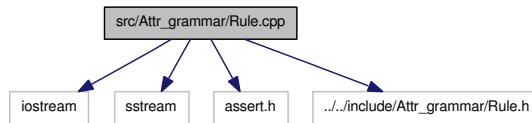
Definition in file [Function.cpp](#).

8.27 src/Attr_grammar/Rule.cpp File Reference

Implementation of the methods the Rule.h.

```
#include <iostream>
#include <sstream>
#include <assert.h>
#include "../../include/Attr_grammar/Rule.h"
```

Include dependency graph for Rule.cpp:



Namespaces

- namespace genevalmag

Variables

- vector< Equation * > genevalmag::index_access_eq

8.27.1 Detailed Description

Implementation of the methods the Rule.h.

Date

05/11/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

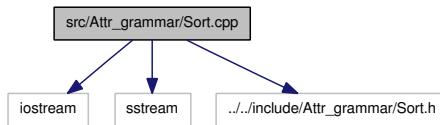
Definition in file Rule.cpp.

8.28 src/Attr_grammar/Sort.cpp File Reference

Implementation of the methods the [Sort.h](#).

```
#include <iostream>
#include <sstream>
#include "../include/Attr_grammar/Sort.h"
```

Include dependency graph for Sort.cpp:



Namespaces

- namespace [genevalmag](#)

8.28.1 Detailed Description

Implementation of the methods the [Sort.h](#).

Date

28/09/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

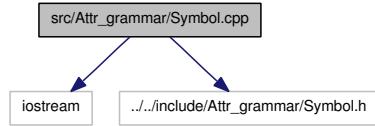
Definition in file [Sort.cpp](#).

8.29 src/Attr_grammar/Symbol.cpp File Reference

Implementation of the methods the [Symbol.h](#).

```
#include <iostream>
#include "../include/Attr_grammar/Symbol.h"
```

Include dependency graph for Symbol.cpp:



Namespaces

- namespace [genevalmag](#)

8.29.1 Detailed Description

Implementation of the methods the [Symbol.h](#).

Date

04/11/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

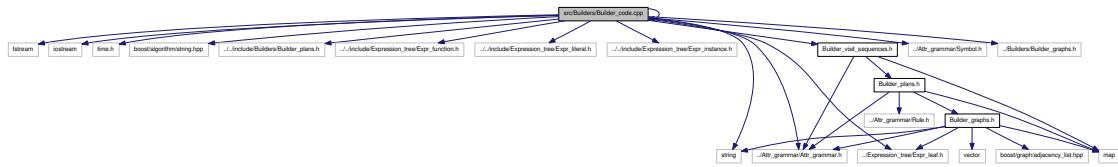
Definition in file [Symbol.cpp](#).

8.30 src/Builders/Builder_code.cpp File Reference

Implementation of the methods the [Builder_code.h](#).

```
#include <fstream>
#include <iostream>
#include <time.h>
#include <boost/algorithm/string.hpp>
#include "../../include/Builders/Builder_plans.h"
#include "../../include/Expression_tree/Expr_function.h"
#include "../../include/Expression_tree/Expr_literal.h"
#include "../../include/Expression_tree/Expr_instance.h"
#include "../../include/Util/Utilities.h"
#include <string>
#include "../Expression_tree/Expr_leaf.h"
#include "../Attr_grammar/Symbol.h"
#include "../Builders/Builder_graphs.h"
#include "../Attr_grammar/Attr_grammar.h"
#include "Builder_visit_sequences.h"
```

Include dependency graph for Builder_code.cpp:



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



Namespaces

- namespace [genevalmag](#)

Functions

- template<class T >
`string genevalmag::write_vector_with_inic (string &text_buffer, const string name_vec, const size_t index, const vector< T > &vec, const string type_vec, const string type_array)`

- string genevalmag::generate_key_plan (string &text, const string &n_key, const int &num_key, const Key_plan &k_p)
- string genevalmag::generate_return_index_context ()
- string genevalmag::generate_expr_text (const Expression *node, const Rule &rule)

8.30.1 Detailed Description

Implementation of the methods the [Builder_code.h](#).

Date

18/03/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

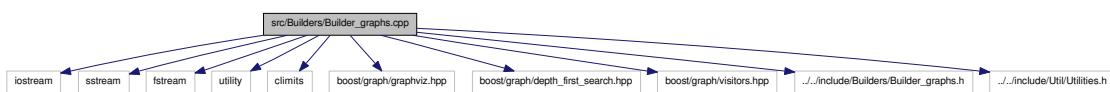
Definition in file [Builder_code.cpp](#).

8.31 src/Builders/Builder_graphs.cpp File Reference

Implementation of the methods the Builder_graph.h.

```
#include <iostream>
#include <sstream>
#include <fstream>
#include <utility>
#include <climits>
#include <boost/graph/graphviz.hpp>
#include <boost/graph/depth_first_search.hpp>
#include <boost/graph/visitors.hpp>
#include "../include/Builders/Builder_graphs.h"
#include "../include/Util/Utilities.h"
```

Include dependency graph for Builder_graphs.cpp:



Classes

- struct genevalmag::cycle_detector

Namespaces

- namespace genevalmag

Functions

- const string genevalmag::PATH_OUTPUT_GRAPHS ("graphs/")
- const string genevalmag::PATH_OUTPUT_DP ("1_DP_graphs/")
- const string genevalmag::PATH_OUTPUT_DOWN ("2_DOWN_graphs/")
- const string genevalmag::PATH_OUTPUT_DCG ("3_DCG_graphs/")
- const string genevalmag::PATH_OUTPUT_ADG ("4_ADG_graphs/")
- const string genevalmag::PATH_OUTPUT_CYCLIC ("CYCLIC_graphs/")
- const string genevalmag::FILE_DP_GRAPH ("_dp_graph")
- const string genevalmag::FILE_DOWN_GRAPH ("_down_graph")
- const string genevalmag::FILE_DCG_GRAPH ("_dcg_graph")
- const string genevalmag::FILE_ADG_GRAPH ("_adg_graph")
- const string genevalmag::FILE_ADG_SUBGRAPH_CYCLIC ("_adg_subgraph_with_cyclic")

8.31.1 Detailed Description

Implementation of the methods the Builder_graph.h.

Date

17/02/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

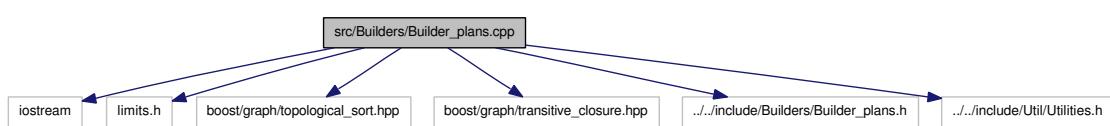
Definition in file [Builder_graphs.cpp](#).

8.32 src/Builders/Builder_plans.cpp File Reference

Implementation of the methods the [Builder_plans.h](#).

```
#include <iostream>
#include <limits.h>
#include <boost/graph/topological_sort.hpp>
#include <boost/graph/transitive_closure.hpp>
#include "../../include/Builders/Builder_plans.h"
#include "../../include/Util/Utilities.h"
```

Include dependency graph for Builder_plans.cpp:



Namespaces

- namespace [genevalmag](#)

Functions

- const string [genevalmag::PATH_OUT_PLAN](#) ("plans/")
- const string [genevalmag::PATH_OUT_PLAN_PROJECT](#) ("plans_project/")
- void [genevalmag::purge_plan_with](#) (const Rule &rule, const Order_eval_eq &order_eq, Order_eval_eq &purged_order)
- bool [genevalmag::defined_work](#) (const vector< Item_work > &list, const Item_work &item_work)
- unsigned short [genevalmag::return_index_vec](#) (const Order_eval_eq &order, vector< Order_eval_eq > &vec)

8.32.1 Detailed Description

Implementation of the methods the [Builder_plans.h](#).

Date

18/02/2010

Author

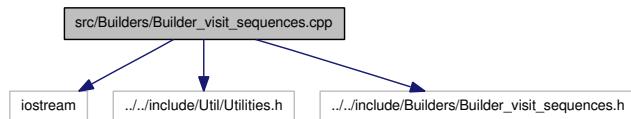
Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

Definition in file [Builder_plans.cpp](#).

8.33 src/Builders/Builder_visit_sequences.cpp File Reference

```
#include <iostream>
#include "../../include/Util/Utilities.h"
#include "../../include/Builders/Builder_visit_sequences.h"
```

Include dependency graph for Builder_visit_sequences.cpp:



Namespaces

- namespace `genevalmag`

Functions

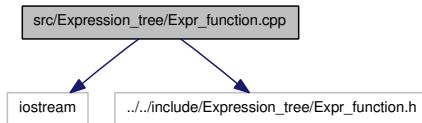
- const unsigned short `genevalmag::LEAVE` (0)
- bool `genevalmag::ins_attr_computed` (const `Expr_instance` *`ins`, const `vector< Expr_instance >` &`vec`)
- void `genevalmag::get_inherits_of` (const `Symbol` *`symb`, const `vector< Expr_instance >` &`computed`, `vector< Expr_instance >` &`rec_child`)
- bool `genevalmag::belong_it` (const `map< Key_plan, unsigned short >`::`const_iterator` `it`, const `vector< map< Key_plan, unsigned short >`::`const_iterator` &`vec`)
- void `genevalmag::merge_vec` (const `vector< map< Key_plan, unsigned short >`::`const_iterator` &`vec_source`, `vector< map< Key_plan, unsigned short >`::`const_iterator` &`vec_targed`)
- void `genevalmag::merge_vec_without_plan` (const `vector< map< Key_plan, unsigned short >`::`const_iterator` &`vec_source`, `vector< map< Key_plan, unsigned short >`::`const_iterator` &`vec_targed`, const `map< Key_plan, unsigned short >`::`const_iterator` &`plan`)
- void `genevalmag::plan_family_computed` (const `vector< map< Key_plan, unsigned short >`::`const_iterator` &`plans_computed`, `vector< unsigned short >` &`visit_seq_computed`)

8.34 src/Expression_tree/Expr_function.cpp File Reference

Implementation of a function element of an Expression.

```
#include <iostream>
#include "../../include/Expression_tree/Expr_function.h"
```

Include dependency graph for Expr_function.cpp:



Namespaces

- namespace [genevalmag](#)

8.34.1 Detailed Description

Implementation of a function element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

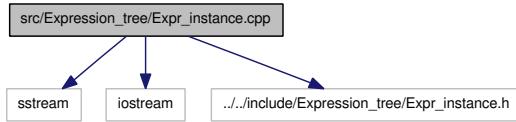
Definition in file [Expr_function.cpp](#).

8.35 src/Expression_tree/Expr_instance.cpp File Reference

Implementation of a instance element of an Expression.

```
#include <sstream>
#include <iostream>
#include "../../include/Expression_tree/Expr_instance.h"
```

Include dependency graph for Expr_instance.cpp:



Namespaces

- namespace [genevalmag](#)

8.35.1 Detailed Description

Implementation of a instance element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

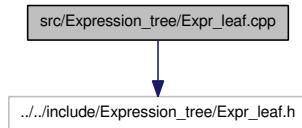
Definition in file [Expr_instance.cpp](#).

8.36 src/Expression_tree/Expr_leaf.cpp File Reference

Implementation element of an Expression.

```
#include "../../../include/Expression_tree/Expr_leaf.h"
```

Include dependency graph for Expr_leaf.cpp:



Namespaces

- namespace `genevalmag`

8.36.1 Detailed Description

Implementation element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

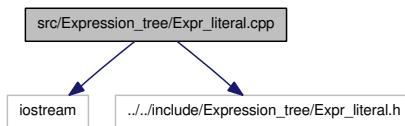
Definition in file [Expr_leaf.cpp](#).

8.37 src/Expression_tree/Expr_literal.cpp File Reference

Implementation of a literal element of an Expression.

```
#include <iostream>
#include "../include/Expression_tree/Expr_literal.h"
```

Include dependency graph for Expr_literal.cpp:



Namespaces

- namespace [genevalmag](#)

8.37.1 Detailed Description

Implementation of a literal element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

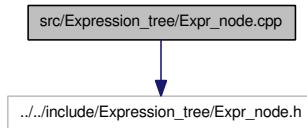
Definition in file [Expr_literal.cpp](#).

8.38 src/Expression_tree/Expr_node.cpp File Reference

Implementation element of an Expression.

```
#include "../../../include/Expression_tree/Expr_node.h"
```

Include dependency graph for Expr_node.cpp:



Namespaces

- namespace [genevalmag](#)

8.38.1 Detailed Description

Implementation element of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

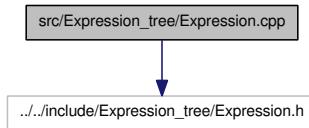
Definition in file [Expr_node.cpp](#).

8.39 src/Expression_tree/Expression.cpp File Reference

Implementation of an Expression.

```
#include "../../../include/Expression_tree/Expression.h"
```

Include dependency graph for Expression.cpp:



Namespaces

- namespace [genevalmag](#)

8.39.1 Detailed Description

Implementation of an Expression.

Date

21/12/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

Definition in file [Expression.cpp](#).

8.40 src/maggen.cpp File Reference

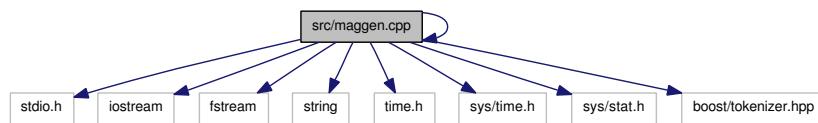
Tool for generation of Static Evaluators for Multiplans Attribute Grammar.

```

#include <stdio.h>
#include <iostream>
#include <fstream>
#include <string>
#include <time.h>
#include <sys/time.h>
#include <sys/stat.h>
#include <boost/tokenizer.hpp>
#include "../include/Maglib.h"

```

Include dependency graph for maggen.cpp:



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



Namespaces

- namespace genevalmag

Functions

- const string genevalmag::DEFAULT_PATH ("./out_maggen/")
- const string genevalmag::DEFAULT_FILE_NAME ("mag_eval")
- const string genevalmag::DEFAULT_INPUT_FILE ("/tmp/.input_maggen_default")
- double genevalmag::timeval_diff (struct timeval *a, struct timeval *b)
- bool genevalmag::check_file_exist (const string &strFilename)
- bool genevalmag::check_name (const string &strFilename)
- void genevalmag::show_help_information ()
- bool genevalmag::parse_parameters (int argc, char *argv[], string &path_input_file, string &path_folder_output, string &name_library, vector< string > &headers)
- int main (int argc, char *argv[], char *envp[])

8.40.1 Detailed Description

Tool for generation of Static Evaluators for Multiplans Attribute Grammar.

Date

09/02/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

Definition in file [maggen.cpp](#).

8.40.2 Function Documentation

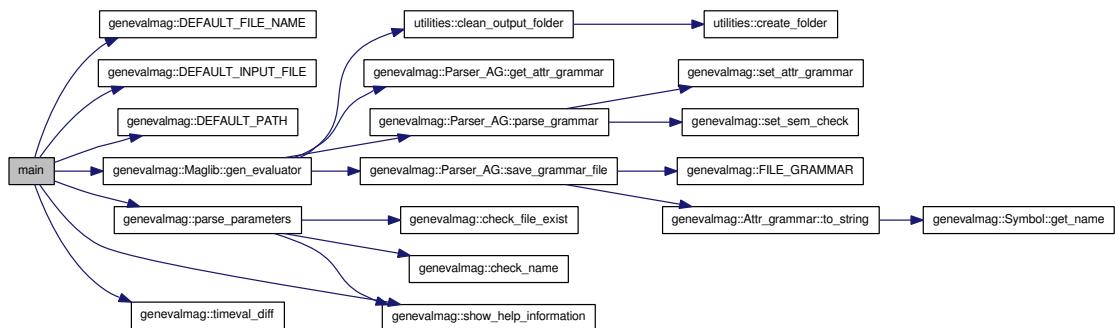
8.40.2.1 int main (int argc, char * argv[], char * envp[])

Main method of the parsing.

Definition at line 234 of file [maggen.cpp](#).

References `genevalmag::DEFAULT_FILE_NAME()`, `genevalmag::DEFAULT_INPUT_FILE()`, `genevalmag::DEFAULT_PATH()`, `genevalmag::Maglib::gen_evaluator()`, `genevalmag::parse_parameters()`, `genevalmag::show_help_information()`, and `genevalmag::timeval_diff()`.

Here is the call graph for this function:



8.41 src/Maglib.cpp File Reference

Implementation of the library of evaluator's generator.

```
#include <vector>
#include <iostream>
#include <string>
#include "../include/Parser/Parser_AG.h"
#include "../include/Builders/Builder_plans.h"
#include "../include/Builders/Builder_visit_sequences.h"
#include "../include/Builders/Builder_code.h"
#include "../include/Util/Utilities.h"
#include "../include/Maglib.h"
```

Include dependency graph for Maglib.cpp:



Namespaces

- namespace `genevalmag`

8.41.1 Detailed Description

Implementation of the library of evaluator's generator.

Date

14/04/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

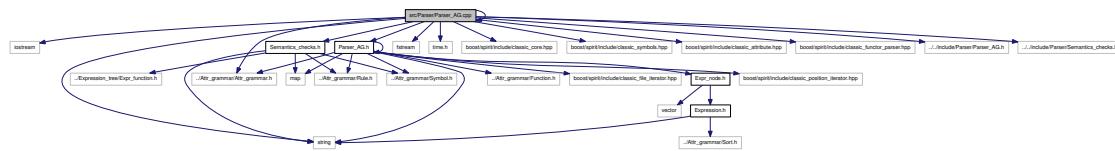
Definition in file [Maglib.cpp](#).

8.42 src/Parser/Parser_AG.cpp File Reference

Header of parsing module of Attribute Grammar.

```
#include <iostream>
#include <string>
#include <fstream>
#include <time.h>
#include <boost/spirit/include/classic_core.hpp>
#include <boost/spirit/include/classic_symbols.hpp>
#include <boost/spirit/include/classic_attribute.hpp>
#include <boost/spirit/include/classic_functor_parser.hpp>
#include " ../../include/Parser/Parser_AG.h"
#include " ../../include/Parser/Semantics_checks.h"
#include " ../../include/Parser/Semantics_actions.h"
#include "../Attr_grammar/Attr_grammar.h"
#include "Semantics_checks.h"
#include "Parser_AG.h"
```

Include dependency graph for Parser_AG.cpp:



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



Classes

- struct genevalmag::skip_parser
- struct genevalmag::skip_parser::definition< ScannerT >
- struct genevalmag::attribute_grammar
- struct genevalmag::attribute_grammar::definition< ScannerT >

Namespaces

- namespace genevalmag

Functions

- const string genevalmag::FILE_GRAMMAR ("Grammar_mag.log")
- std::ostream & genevalmag::operator<< (std::ostream &out, file_position const &lc)

8.42.1 Detailed Description

Header of parsing module of Attribute Grammar. Implementation of parsing module of Attribute Grammar, [Parser_AG.h](#).

Date

09/02/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

Date

13/09/2009

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

Definition in file [Parser_AG.cpp](#).

8.43 src/Parser/Semantics_actions.cpp File Reference

Implementation of the methods the [Semantics_actions.h](#).

```
#include <iostream>
#include <sstream>
#include <boost/algorithm/string/erase.hpp>
#include "../../include/Parser/Semantics_actions.h"
#include "../../include/Expression_tree/Expr_instance.h"
#include "../../include/Expression_tree/Expr_literal.h"
#include "../../include/Expression_tree/Expr_function.h"
#include "../../include/Util/Utilities.h"
```

Include dependency graph for Semantics_actions.cpp:



Classes

- struct [genevalmag::decl_attribute](#)

Namespaces

- namespace [genevalmag](#)

Functions

- void [genevalmag::set_attr_grammar](#) (Attr_grammar *at_grammar)
- void [genevalmag::set_sem_check](#) (Semantics_checks *s_check)
- void [genevalmag::create_sort](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::inic_func](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::add_function](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::save_name_func](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::save_domain_func](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::save_image_func](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::add_operator](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::save_mode_op](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::save_prec_op](#) (int const prec)
- void [genevalmag::save_assoc_op](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::add_attribute](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::save_sort_attr](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::save_type_attr](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::save_member_list_attr](#) (const iterator_t str, const iterator_t end)
- void [genevalmag::create_attributes](#) (const iterator_t str, const iterator_t end)

- void `genevalmag::create_new_non_terminal` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_new_terminal` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_rule` (const iterator_t str, const iterator_t end)
- void `genevalmag::save_right_side_rule` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_abbreviated_rule` (const iterator_t str, const iterator_t end)
- void `genevalmag::save_rule` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_instance` (const iterator_t str, const iterator_t end)
- void `genevalmag::save_index_ins` (int const index)
- void `genevalmag::save_attr_ins` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_lit_number` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_lit_ch` (const iterator_t ch, const iterator_t end)
- void `genevalmag::create_lit_str` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_bool` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_function` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_operator` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_equation` (const iterator_t str, const iterator_t end)
- void `genevalmag::save_rvalue` (const iterator_t str, const iterator_t end)
- void `genevalmag::push_mark` (char name)
- void `genevalmag::create_literal_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_instance_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_func_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_root_infix_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_root_function_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_root_postfix_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::create_root_prefix_node` (const iterator_t str, const iterator_t end)
- void `genevalmag::check_well_defined` (const iterator_t str, const iterator_t end)
- void `genevalmag::increment_level` (char name)
- void `genevalmag::decrement_level` (char name)

Variables

- Attr_grammar * `genevalmag::attr_grammar`
- Semantics_checks * `genevalmag::sem_check`
- Function * `genevalmag::current_func`
- struct `genevalmag::decl_attribute` * `genevalmag::new_attrs`
- Rule * `genevalmag::current_rule`
- Expr_instance * `genevalmag::current_instance`
- Expr_literal * `genevalmag::current_literal`
- Expr_function * `genevalmag::current_ast_function`
- Equation * `genevalmag::current_eq`
- vector< Expression * > `genevalmag::stack_node`
- vector< Expr_node * > `genevalmag::stack_inner_node`

8.43.1 Detailed Description

Implementation of the methods the [Semantics_actions.h](#).

Date

09/02/2010

Author

Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
Picco, Gonzalo Martin <gonzalopicco@gmail.com>

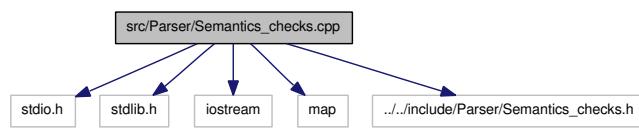
Definition in file [Semantics_actions.cpp](#).

8.44 src/Parser/Semantics_checks.cpp File Reference

Implementation of the methods the [Semantics_checks.h](#).

```
#include <stdio.h>
#include <stdlib.h>
#include <iostream>
#include <map>
#include "../include/Parser/Semantics_checks.h"
```

Include dependency graph for Semantics_checks.cpp:



Namespaces

- namespace [genevalmag](#)

Functions

- int [genevalmag::swap_root_child](#) (Expr_function **old_root, int i_new_root)
- int [genevalmag::swap_root_grandson](#) (Expr_function **old_root)
- void [genevalmag::warshall_algorithm](#) (const unsigned int size, bool *matrix_plain)
- int [genevalmag::get_index](#) (string name_symb, vector< string > non_term)
- bool [genevalmag::check_eqDefines_it](#) (const Symbol *symb, const int index, const Attribute *attr, const map< unsigned short, Equation > eqs)

8.44.1 Detailed Description

Implementation of the methods the [Semantics_checks.h](#).

Date

11/02/2010

Author

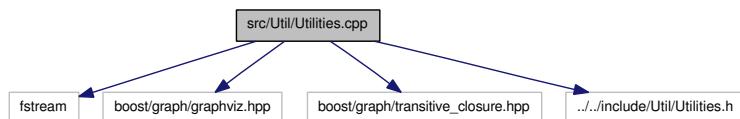
Kilmurray, Gerardo Luis <gerakilmurray@gmail.com>
 Picco, Gonzalo Martin <gonzalopicco@gmail.com>

Definition in file [Semantics_checks.cpp](#).

8.45 src/Util/Utilities.cpp File Reference

```
#include <fstream>
#include <boost/graph/graphviz.hpp>
#include <boost/graph/transitive_closure.hpp>
#include "../include/Util/Utilities.h"
```

Include dependency graph for Utilities.cpp:



Namespaces

- namespace `utilities`

Functions

- bool `utilities::create_folder (const string path)`
- bool `utilities::clean_output_folder (const string path)`
- bool `utilities::copy_static_code (const string path_d, const string path_s)`
- void `utilities::generate_names_instance (const Graph &graph, string datas[], size_t size_d)`
- void `utilities::generate_names_attr (const Graph &graph, string datas[], size_t size_d)`
- void `utilities::print_graph (const Graph &graph, const string path, const string name_file, const string name_graph, const string names[], string shape_vertex)`
- Vertex `utilities::return_vertex (const Graph &graph, const Expr_leaf *node)`
- void `utilities::merge_graph (const Graph &graph1, const Graph &graph2, Graph &graph_merged)`
- void `utilities::project_graph (const Symbol *symb, Graph &graph)`
- bool `utilities::belong_index (const unsigned short &index, const vector< unsigned short > &vec)`
- string `utilities::cleaning_tabs (const string str)`
- string `utilities::write_inf_context (const vector< unsigned short > &context_vec)`

Index

~Attr_grammar
 genevalmag::Attr_grammar, 65
~Attribute
 genevalmag::Attribute, 82
~Builder_code
 genevalmag::Builder_code, 108
~Builder_graphs
 genevalmag::Builder_graphs, 124
~Builder_plans
 genevalmag::Builder_plans, 140
~Builder_visit_sequences
 genevalmag::Builder_visit_sequences,
 158
~Equation
 genevalmag::Equation, 172
~Expr_function
 genevalmag::Expr_function, 181
~Expr_instance
 genevalmag::Expr_instance, 194
~Expr_leaf
 genevalmag::Expr_leaf, 205
~Expr_literal
 genevalmag::Expr_literal, 210
~Expr_node
 genevalmag::Expr_node, 218
~Expression
 genevalmag::Expression, 224
~Function
 genevalmag::Function, 231
~Maglib
 genevalmag::Maglib, 255
~Parser_AG
 genevalmag::Parser_AG, 258
~Rule
 genevalmag::Rule, 263
~Semantics_checks
 genevalmag::Semantics_checks, 276
~Sort
 genevalmag::Sort, 289
~Symbol
 genevalmag::Symbol, 296
_get_count_ref
 genevalmag::Equation, 172
_graph_cycle
 genevalmag::cycle_detector, 167
_has_cycle
 genevalmag::cycle_detector, 167
a_member_symbol
 genevalmag::Attribute, 89
a_mod_type
 genevalmag::Attribute, 89
a_name
 genevalmag::Attribute, 90
a_sort_type
 genevalmag::Attribute, 90
active_conflict
 genevalmag::Expression, 224
add
 genevalmag, 17
add_attr
 genevalmag::Symbol, 297
add_attribute
 genevalmag, 17
 genevalmag::Attr_grammar, 66
add_child
 genevalmag::Expr_node, 218
add_domain
 genevalmag::Function, 232
add_eq
 genevalmag::Rule, 263
add_function
 genevalmag, 18
 genevalmag::Attr_grammar, 66
add_operator
 genevalmag, 18
add_right_symbol
 genevalmag::Rule, 264
add_rule
 genevalmag::Attr_grammar, 67
add_sort
 genevalmag::Attr_grammar, 67
add_symbol
 genevalmag::Attr_grammar, 68
ag_attr
 genevalmag::Attr_grammar, 78
ag_func
 genevalmag::Attr_grammar, 78
ag_initial_symb

genevalmag::Attr_grammar, 78
 ag_rule
 genevalmag::Attr_grammar, 78
 ag_sort
 genevalmag::Attr_grammar, 78
 ag_symb_non_terminals
 genevalmag::Attr_grammar, 79
 ag_symb_terminals
 genevalmag::Attr_grammar, 79
 all_visit_seqs
 genevalmag::Builder_visit_sequences,
 162
 Attr_grammar
 genevalmag::Attr_grammar, 65
 attr_grammar
 genevalmag, 52
 genevalmag::Builder_code, 120
 genevalmag::Builder_graphs, 136
 genevalmag::Builder_plans, 154
 genevalmag::Builder_visit_sequences,
 162
 genevalmag::Parser_AG, 260
 attr_vertex_graphs
 genevalmag::Builder_graphs, 136
 Attribute
 genevalmag::Attribute, 81

 b_plans
 genevalmag::Builder_code, 120
 genevalmag::Builder_visit_sequences,
 162
 b_v_seq
 genevalmag::Builder_code, 120
 back_edge
 genevalmag::cycle_detector, 166
 belong
 genevalmag, 19
 belong_index
 utilities, 54
 belong_it
 genevalmag, 19
 belongs_non_terminal
 genevalmag::Rule, 264
 build_graphs
 genevalmag::Builder_plans, 154
 build_plans
 genevalmag::Builder_plans, 140
 Builder_code
 genevalmag::Builder_code, 108
 Builder_graphs
 genevalmag::Builder_graphs, 124
 Builder_plans
 genevalmag::Builder_plans, 140

 Builder_visit_sequences
 genevalmag::Builder_visit_sequences,
 158

 char_t
 genevalmag, 15
 check_all_defined_non_terminal
 genevalmag::Semantics_checks, 277
 check_cyclic_adp_dependencies
 genevalmag::Builder_graphs, 124
 check_eqDefines_it
 genevalmag, 20
 check_fileExist
 genevalmag, 20
 check_name
 genevalmag, 20
 check_reachability
 genevalmag::Semantics_checks, 277
 check_wellDefined
 genevalmag, 20
 check_wellDefined_AG
 genevalmag::Semantics_checks, 278
 childs
 genevalmag::Expr_node, 220
 clean_output_folder
 utilities, 54
 cleaning_tabs
 utilities, 55
 combined_inf_contexts
 genevalmag::Builder_graphs, 124
 compare_order
 genevalmag::Expr_function, 181
 compare_precedence
 genevalmag::Expr_function, 182
 complete_dp_graphs
 genevalmag::Builder_graphs, 125
 compute_adp_graph
 genevalmag::Builder_graphs, 126
 compute_attr_vertex
 genevalmag::Builder_graphs, 126
 compute_dcg
 genevalmag::Builder_graphs, 127
 compute_dependency_graphs
 genevalmag::Builder_graphs, 128
 compute_down_graph
 genevalmag::Builder_graphs, 128
 compute_order
 genevalmag::Builder_plans, 141
 conflict
 genevalmag::Expression, 228
 context
 genevalmag::c_rule, 165
 Context_rule

genevalmag, 15
contexts_uniques
 genevalmag::Builder_plans, 154
copy
 genevalmag::Attribute, 82
 genevalmag::Equation, 172
 genevalmag::Expr_function, 182
 genevalmag::Expr_instance, 195
 genevalmag::Expr_literal, 210
 genevalmag::Function, 232
 genevalmag::Rule, 265
 genevalmag::Sort, 290
 genevalmag::Symbol, 297
copy_static_code
 utilities, 56
correct_associativity
 genevalmag::Semantics_checks, 279
correct_precedence
 genevalmag::Semantics_checks, 279
correct_subtree
 genevalmag::Semantics_checks, 280
count_eqs
 genevalmag::Attr_grammar, 79
count_non_terminal
 genevalmag::Rule, 265
count_ref
 genevalmag::Equation, 176
create_abbreviated_rule
 genevalmag, 21
create_attributes
 genevalmag, 21
create_bool
 genevalmag, 22
create_equation
 genevalmag, 23
create_folder
 utilities, 56
create_func_node
 genevalmag, 23
create_function
 genevalmag, 24
create_instance
 genevalmag, 24
create_instance_node
 genevalmag, 25
create_lit_ch
 genevalmag, 25
create_lit_number
 genevalmag, 26
create_lit_str
 genevalmag, 26
create_literal_node
 genevalmag, 26
create_new_non_terminal
 genevalmag, 27
create_new_terminal
 genevalmag, 27
create_operator
 genevalmag, 28
create_root_function_node
 genevalmag, 28
create_root_infix_node
 genevalmag, 28
create_root_postfix_node
 genevalmag, 29
create_root_prefix_node
 genevalmag, 30
create_rule
 genevalmag, 30
create_sort
 genevalmag, 31
current_ast_function
 genevalmag, 52
current_eq
 genevalmag, 52
current_func
 genevalmag, 52
current_instance
 genevalmag, 52
current_literal
 genevalmag, 52
current_rule
 genevalmag, 52
cycle_detector
 genevalmag::cycle_detector, 166
d_member_symbol
 genevalmag::decl_attribute, 168
d_mod_type
 genevalmag::decl_attribute, 168
d_names
 genevalmag::decl_attribute, 168
d_sort_type
 genevalmag::decl_attribute, 168
decrement_level
 genevalmag, 31
decrement_precedence_level
 genevalmag::Semantics_checks, 281
DEFAULT_FILE_NAME
 genevalmag, 31
DEFAULT_INPUT_FILE
 genevalmag, 32
DEFAULT_PATH
 genevalmag, 32
defined_equation
 genevalmag::Rule, 266

defined_rule
 genevalmag::Attr_grammar, 68

defined_work
 genevalmag, 32

definition
 genevalmag::attribute_-
 grammar::definition, 93

 genevalmag::skip_parser::definition, 286

delete_parent
 genevalmag::Expression, 224

desactive_conflict
 genevalmag::Expression, 225

destroy
 genevalmag::Attribute, 83

 genevalmag::Equation, 172

 genevalmag::Expr_function, 183

 genevalmag::Expr_instance, 195

 genevalmag::Expr_literal, 211

 genevalmag::Function, 233

 genevalmag::Rule, 267

 genevalmag::Sort, 290

 genevalmag::Symbol, 298

eq_id
 genevalmag::Equation, 177

equals
 genevalmag::Attribute, 83

 genevalmag::Equation, 172

 genevalmag::Expr_leaf, 205

 genevalmag::Function, 233

 genevalmag::Rule, 267

 genevalmag::Sort, 290

 genevalmag::Symbol, 298

equals_without_index
 genevalmag::Expr_instance, 196

Equation
 genevalmag::Equation, 171

eval_plans
 genevalmag::Builder_plans, 154

examine_edge
 genevalmag::cycle_detector, 166

Expr_function
 genevalmag::Expr_function, 180

Expr_instance
 genevalmag::Expr_instance, 193

Expr_leaf
 genevalmag::Expr_leaf, 205

Expr_literal
 genevalmag::Expr_literal, 209

Expr_node
 genevalmag::Expr_node, 218

Expression
 genevalmag::Expression, 224

f_assoc
 genevalmag::Function, 243

f_domain
 genevalmag::Function, 243

f_image
 genevalmag::Function, 243

f_is_operator
 genevalmag::Function, 243

f_mod
 genevalmag::Function, 243

f_name
 genevalmag::Function, 243

f_prec
 genevalmag::Function, 243

father
 genevalmag::c_rule, 165

 genevalmag::k_w, 254

FILE_ADP_GRAPH
 genevalmag, 32

FILE_ADP_SUBGRAPH_CYCLIC
 genevalmag, 33

FILE_DCG_GRAPH
 genevalmag, 33

FILE_DOWN_GRAPH
 genevalmag, 33

FILE_DP_GRAPH
 genevalmag, 33

FILE_GRAMMAR
 genevalmag, 33

file_name
 genevalmag::Builder_code, 121

func
 genevalmag::Expr_function, 190

Function
 genevalmag::Function, 231

gen_evaluator
 genevalmag::Maglib, 255

gen_visit_seq
 genevalmag::Builder_visit_sequences,
 159

generate_add_plan
 genevalmag::Builder_code, 108

generate_add_plan_project
 genevalmag::Builder_code, 109

generate_all_compute_eq
 genevalmag::Builder_code, 109

generate_all_methods_eqs
 genevalmag::Builder_code, 110

generate_code
 genevalmag::Builder_code, 111

generate_code_file
 genevalmag::Builder_code, 111

generate_constructor
 genevalmag::Builder_code, 112
generate_eval_visiter
 genevalmag::Builder_code, 112
generate_evaluator
 genevalmag::Builder_code, 113
generate_expr_text
 genevalmag, 34
generate_footer_code
 genevalmag::Builder_code, 113
generate_footer_header
 genevalmag::Builder_code, 114
generate_graphs
 genevalmag::Builder_plans, 142
generate_header_file
 genevalmag::Builder_code, 114
generate_initialize_context
 genevalmag::Builder_code, 114
generate_initialize_plan_proj
 genevalmag::Builder_code, 115
generate_initialize_plans
 genevalmag::Builder_code, 115
generate_initialize_rules
 genevalmag::Builder_code, 116
generate_initialize_v_seq
 genevalmag::Builder_code, 116
generate_key_plan
 genevalmag, 35
generate_methods
 genevalmag::Builder_code, 117
generate_names_attr
 utilities, 57
generate_names_instance
 utilities, 57
generate_plans
 genevalmag::Builder_plans, 143
generate_print
 genevalmag::Builder_code, 118
generate_return_index_context
 genevalmag, 36
generate_structs
 genevalmag::Builder_code, 118
generate_translate
 genevalmag::Builder_code, 119
generate_traverse
 genevalmag::Builder_code, 120
generate_visit_sequences
 genevalmag::Builder_visit_sequences,
 160
generates_topological_order
 genevalmag::Builder_plans, 144
genevalmag, 11
 add, 17
 add_attribute, 17
 add_function, 18
 add_operator, 18
 attr_grammar, 52
 belong, 19
 belong_it, 19
 char_t, 15
 check_eqDefines_it, 20
 check_file_exist, 20
 check_name, 20
 check_well_defined, 20
 Context_rule, 15
 create_abbreviated_rule, 21
 create_attributes, 21
 create_bool, 22
 create_equation, 23
 create_func_node, 23
 create_function, 24
 create_instance, 24
 create_instance_node, 25
 create_lit_ch, 25
 create_lit_number, 26
 create_lit_str, 26
 create_literal_node, 26
 create_new_non_terminal, 27
 create_new_terminal, 27
 create_operator, 28
 create_root_function_node, 28
 create_root_infix_node, 28
 create_root_postfix_node, 29
 create_root_prefix_node, 30
 create_rule, 30
 create_sort, 31
 current_ast_function, 52
 current_eq, 52
 current_func, 52
 current_instance, 52
 current_literal, 52
 current_rule, 52
 decrement_level, 31
 DEFAULT_FILE_NAME, 31
 DEFAULT_INPUT_FILE, 32
 DEFAULT_PATH, 32
 defined_work, 32
 FILE_AD_P_GRAPH, 32
 FILE_AD_P_SUBGRAPH_CYCLIC, 33
 FILE_DCG_GRAPH, 33
 FILE_DOWN_GRAPH, 33
 FILE_DP_GRAPH, 33
 FILE_GRAMMAR, 33
 generate_expr_text, 34
 generate_key_plan, 35
 generate_return_index_context, 36

get_index, 36
 get_inherits_of, 36
 Graph, 15
 increment_level, 36
 index_access_eq, 53
 inic_func, 37
 ins_attr_computed, 37
 IS_OPERATOR, 37
 Item_work, 15
 iterator_f, 15
 iterator_t, 15
 k_bool, 16
 k_char, 16
 k_float, 16
 k_infix, 17
 k_inherit, 17
 k_int, 16
 k_left, 16
 k_non_assoc, 16
 k_non_terminal, 17
 k_postfix, 17
 k_prefix, 17
 k_right, 16
 k_string, 16
 k_synthetize, 17
 k_terminal, 17
 Key_plan, 15
 Key_plan_project, 15
 Key_work_list, 15
 LEAVE, 38
 literal_type, 16
 merge_vec, 38
 merge_vec_without_plan, 38
 newAttrs, 53
 oper_assoc, 16
 oper_mode, 16
 operator<<, 39
 Order_eval_eq, 15
 Order_rule, 16
 parse_parameters, 39
 PATH_OUT_PLAN, 39
 PATH_OUT_PLAN_PROJECT, 40
 PATH_OUTPUT_ADP, 40
 PATH_OUTPUT_CYCLIC, 40
 PATH_OUTPUT_DCG, 40
 PATH_OUTPUT_DOWN, 40
 PATH_OUTPUT_DP, 41
 PATH_OUTPUT_GRAPHS, 41
 plan_family_computed, 41
 purge_plan_with, 42
 push_mark, 42
 return_index_vec, 42
 save_assoc_op, 43
 save_attr_ins, 43
 save_domain_func, 44
 save_image_func, 44
 save_index_ins, 45
 save_member_list_attr, 45
 save_mode_op, 46
 save_name_func, 46
 save_prec_op, 46
 save_right_side_rule, 47
 save_rule, 47
 save_rvalue, 47
 save_sort_attr, 48
 save_type_attr, 49
 sem_check, 53
 set_attr_grammar, 49
 set_sem_check, 49
 show_help_information, 50
 stack_inner_node, 53
 stack_node, 53
 swap_root_child, 50
 swap_root_grandson, 50
 symbol_type, 17
 timeval_diff, 51
 to_string_map, 51
 type_attr, 17
 Vertex, 16
 Visit_seq, 16
 warshall_algorithm, 51
 write_vector_with_inic, 51
 genevalmag::Attr_grammar, 63
 ~Attr_grammar, 65
 add_attribute, 66
 add_function, 66
 add_rule, 67
 add_sort, 67
 add_symbol, 68
 ag_attr, 78
 ag_func, 78
 ag_initial_symb, 78
 ag_rule, 78
 ag_sort, 78
 ag_symb_non_terminals, 79
 ag_symb_terminals, 79
 Attr_grammar, 65
 count_eqs, 79
 defined_rule, 68
 get_count_eqs, 69
 get_eq, 69
 get_eq_l_value, 70
 get_function, 70
 get_functions, 71
 get_index_eq_with_context, 71
 get_initial_symb, 72

get_non_terminal_symbols, 72
get_rule, 73
get_rules, 73
get_rules_with_left_symbol, 74
get_sorts, 75
get_symbol, 75
load_attributes, 76
return_sort, 76
to_string, 77
genevalmag::Attribute, 80
 ~Attribute, 82
 a_member_symbol, 89
 a_mod_type, 89
 a_name, 90
 a_sort_type, 90
 Attribute, 81
 copy, 82
 destroy, 83
 equals, 83
 get_member_symbol, 84
 get_mod_type, 84
 get_name, 85
 get_sort_type, 85
 is_inherit, 86
 is_synthetize, 86
 key, 86
 operator=, 87
 set_member_symbol, 87
 set_mod_type, 88
 set_name, 88
 set_sort_type, 89
 to_string, 89
genevalmag::attribute_grammar, 91
genevalmag::attribute_grammar::definition,
 92
 definition, 93
 r_att_grammar, 97
 r_attribute_st, 97
 r_attributes, 97
 r_bloq_sem, 97
 r_boolean, 97
 r_char, 97
 r_compute_eq, 98
 r_conj_symb, 98
 r_cpp_basic_types, 98
 r_cpp_reserved_words, 98
 r_decl_attr, 98
 r_decl_func, 98
 r_decl_oper, 98
 r_decl_rule, 98
 r_decl_sort, 99
 r_dom_func, 99
 r_equation, 99
 r_esc_seq, 99
 r_expr_prime, 99
 r_expr_prime_prime, 99
 r_expression, 99
 r_function, 100
 r_function_st, 100
 r_id_op, 100
 r_ident, 100
 r_instance, 100
 r_literal, 100
 r_non_term_st, 100
 r_op_infix_st, 101
 r_op_postfix_st, 101
 r_op_prefix_st, 101
 r_oper, 101
 r_oper_assoc, 101
 r_oper_infix, 101
 r_oper_mode, 101
 r_oper_postfix, 102
 r_oper_prefix, 102
 r_reserved_word, 102
 r_right_rule, 102
 r_rules, 102
 r_semantic_domain, 102
 r_sort_st, 102
 r_sort_stable, 103
 r_string, 103
 r_string_lit, 103
 r_terminal, 103
 r_type_attr, 103
 rule_exp, 93
 rule_lexeme, 93
 st_attributes, 103
 st_functions, 103
 st_non_terminal, 104
 st_op_infix, 104
 st_op_postfix, 104
 st_op_prefix, 104
 st_sorts, 104
 start, 97
genevalmag::Builder_code, 105
 ~Builder_code, 108
 attr_grammar, 120
 b_plans, 120
 b_v_seq, 120
 Builder_code, 108
 file_name, 121
 generate_add_plan, 108
 generate_add_plan_project, 109
 generate_all_compute_eq, 109
 generate_all_methods_eqs, 110
 generate_code, 111
 generate_code_file, 111

generate_constructor, 112
 generate_eval_visiter, 112
 generate_evaluator, 113
 generate_footer_code, 113
 generate_footer_header, 114
 generate_header_file, 114
 generate_initialize_context, 114
 generate_initialize_plan_proj, 115
 generate_initialize_plans, 115
 generate_initialize_rules, 116
 generate_initialize_v_seq, 116
 generate_methods, 117
 generate_print, 118
 generate_structs, 118
 generate_translate, 119
 generate_traverse, 120
 path_output, 121
genevalmag::Builder_graphs, 122
 ~Builder_graphs, 124
 attr_grammar, 136
 attr_vertex_graphs, 136
Builder_graphs, 124
 check_cyclic_adp_dependencies, 124
 combined_inf_contexts, 124
 complete_dp_graphs, 125
 compute_adp_graph, 126
 compute_attr_vertex, 126
 compute_dcg, 127
 compute_dependency_graphs, 128
 compute_down_graph, 128
 get_adp_graphs, 129
 get_cyclic_graphs, 129
 get_dcg_graph, 130
 p_Adp_graphs, 136
 p_Adp_subgraphs_cyclics, 136
 p_Deg_graphs, 137
 p_Down_graphs, 137
 p_Dp_graphs, 137
 save_adp_graphs, 130
 save_all_graphs, 131
 save_cyclic_graphs, 132
 save_dcg_graphs, 133
 save_down_graphs, 134
 save_dp_graphs, 135
genevalmag::Builder_plans, 138
 ~Builder_plans, 140
 attr_grammar, 154
 build_graphs, 154
 build_plans, 140
Builder_plans, 140
 compute_order, 141
 contexts_uniques, 154
 eval_plans, 154
 generate_graphs, 142
 generate_plans, 143
 generates_topological_order, 144
 get_context_unique, 145
 get_contexts_uniques, 145
 get_index_plan, 146
 get_init_order, 146
 get_plan_project, 146
 get_plans, 147
 get_plans_project, 147
 get_plans_project_uniques, 148
 get_plans_uniques, 148
 init_order_ag, 154
 plans_project, 154
 plans_project_uniques, 154
 plans_uniques, 155
 return_index_context, 148
 return_index_plan, 149
 return_index_plan_p, 149
 save_all_graphs, 150
 save_all_plans, 151
 save_all_plans_project, 152
 save_cyclic_graphs, 153
genevalmag::Builder_visit_sequences, 156
 ~Builder_visit_sequences, 158
 all_visit_seqs, 162
 attr_grammar, 162
 b_plans, 162
 Builder_visit_sequences, 158
 gen_visit_seq, 159
 generate_visit_sequences, 160
 get_visit_seq, 161
 print_all_visit_sequences, 162
 update_visit_sequence, 162
genevalmag::c_rule, 164
 context, 165
 father, 165
 operator<, 164
 operator=, 164
genevalmag::cycle_detector, 166
 _graph_cycle, 167
 _has_cycle, 167
 back_edge, 166
 cycle_detector, 166
 examine_edge, 166
genevalmag::decl_attribute, 168
 d_member_symbol, 168
 d_mod_type, 168
 d_names, 168
 d_sort_type, 168
genevalmag::Equation, 169
 ~Equation, 172
 _get_count_ref, 172

copy, 172
count_ref, 176
destroy, 172
eq_id, 177
equals, 172
Equation, 171
get_id, 173
get_instance_right_side, 173
get_l_value, 173
get_r_value, 174
inorder_only_instance, 174
inorder_only_leaf, 174
key, 174
l_value, 177
operator=, 175
r_value, 177
set_id, 175
set_l_value, 175
set_r_value, 176
to_string, 176
genevalmag::Expr_function, 178
~Expr_function, 181
compare_order, 181
compare_precedence, 182
copy, 182
destroy, 183
Expr_function, 180
func, 190
get_function, 183
get_precedence_level, 184
get_syntax_order, 184
is_comparable, 185
is_infix, 185
is_postfix, 186
is_prefix, 186
operator=, 187
precedence_level, 190
set_function, 188
set_precedence_level, 188
set_syntax_order, 188
syntax_order, 190
to_string, 189
genevalmag::Expr_instance, 191
~Expr_instance, 194
copy, 195
destroy, 195
equals_without_index, 196
Expr_instance, 193
get_attr, 196
get_num, 197
get_symb, 197
i_attr, 201
i_num, 201
i_symb, 201
key, 198
operator=, 198
set_attr, 199
set_num, 199
set_symb, 200
to_string, 200
genevalmag::Expr_leaf, 203
~Expr_leaf, 205
equals, 205
Expr_leaf, 205
to_string, 206
genevalmag::Expr_literal, 207
~Expr_literal, 210
copy, 210
destroy, 211
Expr_literal, 209
get_type, 211
get_value, 212
operator=, 212
set_type, 213
set_value, 213
to_string, 214
type, 214
value, 215
genevalmag::Expr_node, 216
~Expr_node, 218
add_child, 218
childs, 220
Expr_node, 218
get_child, 218
get_childs, 219
replace_child, 219
to_string, 220
genevalmag::Expression, 221
~Expression, 224
active_conflict, 224
conflict, 228
delete_parent, 224
desactive_conflict, 225
Expression, 224
get_conflict, 225
get_parent, 225
get_type_synthesized, 226
parent, 228
set_parent, 226
set_type_synthesized, 227
to_string, 227
type_synthesized, 228
genevalmag::Function, 229
~Function, 231
add_domain, 232
copy, 232

destroy, 233
 equals, 233
 f_assoc, 243
 f_domain, 243
 f_image, 243
 f_is_operator, 243
 f_mod, 243
 f_name, 243
 f_prec, 243
 Function, 231
 get_arity, 234
 get_domain, 234
 get_image, 234
 get_mode, 235
 get_name, 235
 get_oper_assoc, 236
 get_prec, 236
 is_operator, 237
 key, 237
 operator=, 238
 set_image, 239
 set_is_operator, 239
 set_mode, 240
 set_name, 240
 set_oper_assoc, 240
 set_prec, 241
 signature, 241
 to_string, 242
 genevalmag::i_w, 245
 item, 246
 operator=, 246
 operator==, 246
 order_attr, 246
 genevalmag::k_p_project, 248
 id_plan_project, 250
 index_occurrence, 250
 operator<, 249
 operator=, 249
 operator==, 249
 symbol_project, 250
 genevalmag::k_plan, 251
 id_plan, 252
 operator<, 251
 operator=, 251
 operator==, 252
 plan, 252
 genevalmag::k_w, 253
 father, 254
 id_rule, 254
 operator=, 253
 operator==, 253
 genevalmag::Maglib, 255
 ~Maglib, 255

gen_evaluator, 255
 Maglib, 255
 genevalmag::Parser_AG, 257
 ~Parser_AG, 258
 attr_grammar, 260
 get_attr_grammar, 258
 parse_grammar, 258
 Parser_AG, 258
 save_grammar_file, 259
 sem_check, 260
 genevalmag::Rule, 261
 ~Rule, 263
 add_eq, 263
 add_right_symbol, 264
 belongs_non_terminal, 264
 copy, 265
 count_non_terminal, 265
 defined_equation, 266
 destroy, 267
 equals, 267
 get_eq, 267
 get_eqs, 268
 get_left_symbol, 268
 get_non_terminals_right_side, 269
 get_number_eqs, 270
 get_offset, 270
 get_right_side, 270
 key, 271
 operator=, 271
 r_eqs, 274
 r_id, 275
 r_left_symbol, 275
 r_offset, 275
 r_right_side, 275
 Rule, 262
 set_id, 272
 set_left_symbol, 272
 to_string, 273
 to_string_not_eqs, 273
 genevalmag::Semantics_checks, 276
 ~Semantics_checks, 276
 check_all_defined_non_terminal, 277
 check_reachability, 277
 check_well_defined_AG, 278
 correct_associativity, 279
 correct_precedence, 279
 correct_subtree, 280
 decrement_precedence_level, 281
 get_index_syntax_order, 281
 get_precedence_level, 282
 increment_index_syntax_order, 282
 increment_precedence_level, 283
 index_syntax_order, 283

precedence_level, 283
reset_semantic_context, 283
Semantics_checks, 276
genevalmag::skip_parser, 285
genevalmag::skip_parser::definition, 286
 definition, 286
 skip, 286
 start, 286
genevalmag::Sort, 288
 ~Sort, 289
 copy, 290
 destroy, 290
 equals, 290
 get_name, 291
 is_type_basic, 291
 key, 292
 operator=, 292
 s_ins, 294
 s_name, 294
 s_type_basic, 294
 set_name, 293
 set_type_basic, 293
 Sort, 288, 289
 to_string, 293
genevalmag::Symbol, 295
 ~Symbol, 296
 add_attr, 297
 copy, 297
 destroy, 298
 equals, 298
 get_attribute, 299
 getAttrs, 300
 get_id, 300
 get_name, 300
 get_type, 301
 is_non_terminal, 302
 key, 302
 operator=, 303
 set_id, 303
 set_name, 303
 set_type, 304
 symbAttrs, 305
 symb_id, 305
 symb_name, 305
 symb_type, 305
 Symbol, 296
 to_string, 304
genevalmag::vertex_data_t, 306
 kind, 306
get_adp_graphs
 genevalmag::Builder_graphs, 129
get_arity
 genevalmag::Function, 234
get_attr
 genevalmag::Expr_instance, 196
get_attr_grammar
 genevalmag::Parser_AG, 258
get_attribute
 genevalmag::Symbol, 299
getAttrs
 genevalmag::Symbol, 300
get_child
 genevalmag::Expr_node, 218
get_childs
 genevalmag::Expr_node, 219
get_conflict
 genevalmag::Expression, 225
get_context_unique
 genevalmag::Builder_plans, 145
get_contexts_uniques
 genevalmag::Builder_plans, 145
get_count_eqs
 genevalmag::Attr_grammar, 69
get_cyclic_graphs
 genevalmag::Builder_graphs, 129
get_dcg_graph
 genevalmag::Builder_graphs, 130
get_domain
 genevalmag::Function, 234
get_eq
 genevalmag::Attr_grammar, 69
 genevalmag::Rule, 267
get_eq_1_value
 genevalmag::Attr_grammar, 70
get_eqs
 genevalmag::Rule, 268
get_function
 genevalmag::Attr_grammar, 70
 genevalmag::Expr_function, 183
get_functions
 genevalmag::Attr_grammar, 71
get_id
 genevalmag::Equation, 173
 genevalmag::Symbol, 300
get_image
 genevalmag::Function, 234
get_index
 genevalmag, 36
get_index_eq_with_context
 genevalmag::Attr_grammar, 71
get_index_plan
 genevalmag::Builder_plans, 146
get_index_syntax_order
 genevalmag::Semantics_checks, 281
get_inherits_of
 genevalmag, 36

get_init_order
 genevalmag::Builder_plans, 146
get_initial_symb
 genevalmag::Attr_grammar, 72
get_instance_right_side
 genevalmag::Equation, 173
get_l_value
 genevalmag::Equation, 173
get_left_symbol
 genevalmag::Rule, 268
get_member_symbol
 genevalmag::Attribute, 84
get_mod_type
 genevalmag::Attribute, 84
get_mode
 genevalmag::Function, 235
get_name
 genevalmag::Attribute, 85
 genevalmag::Function, 235
 genevalmag::Sort, 291
 genevalmag::Symbol, 300
get_non_terminal_symbols
 genevalmag::Attr_grammar, 72
get_non_terminals_right_side
 genevalmag::Rule, 269
get_num
 genevalmag::Expr_instance, 197
get_number_eqs
 genevalmag::Rule, 270
get_offset
 genevalmag::Rule, 270
get_oper_assoc
 genevalmag::Function, 236
get_parent
 genevalmag::Expression, 225
get_plan_project
 genevalmag::Builder_plans, 146
get_plans
 genevalmag::Builder_plans, 147
get_plans_project
 genevalmag::Builder_plans, 147
get_plans_project_uniques
 genevalmag::Builder_plans, 148
get_plans_uniques
 genevalmag::Builder_plans, 148
get_prec
 genevalmag::Function, 236
get_precedence_level
 genevalmag::Expr_function, 184
 genevalmag::Semantics_checks, 282
get_r_value
 genevalmag::Equation, 174
get_right_side
 genevalmag::Rule, 270
get_rule
 genevalmag::Attr_grammar, 73
get_rules
 genevalmag::Attr_grammar, 73
get_rules_with_left_symbol
 genevalmag::Attr_grammar, 74
get_sort_type
 genevalmag::Attribute, 85
get_sorts
 genevalmag::Attr_grammar, 75
get_symb
 genevalmag::Expr_instance, 197
get_symbol
 genevalmag::Attr_grammar, 75
get_syntax_order
 genevalmag::Expr_function, 184
get_type
 genevalmag::Expr_literal, 211
 genevalmag::Symbol, 301
get_type_synthesized
 genevalmag::Expression, 226
get_value
 genevalmag::Expr_literal, 212
get_visit_seq
 genevalmag::Builder_visit_sequences,
 161
Graph
 genevalmag, 15
i_attr
 genevalmag::Expr_instance, 201
i_num
 genevalmag::Expr_instance, 201
i_symb
 genevalmag::Expr_instance, 201
id_plan
 genevalmag::k_plan, 252
id_plan_project
 genevalmag::k_p_project, 250
id_rule
 genevalmag::k_w, 254
include/Attr_grammar/Attr_grammar.h, 307
include/Attr_grammar/Attribute.h, 309
include/Attr_grammar/Equation.h, 311
include/Attr_grammar/Function.h, 312
include/Attr_grammar/Rule.h, 314
include/Attr_grammar/Sort.h, 316
include/Attr_grammar/Symbol.h, 317
include/Builders/Builder_code.h, 319
include/Builders/Builder_graphs.h, 321
include/Builders/Builder_plans.h, 323
include/Builders/Builder_visit_sequences.h,
 325

include/Expression_tree/Expr_function.h,
 327
include/Expression_tree/Expr_instance.h,
 328
include/Expression_tree/Expr_leaf.h, **329**
include/Expression_tree/Expr_literal.h, **330**
include/Expression_tree/Expr_node.h, **331**
include/Expression_tree/Expression.h, **332**
include/Maglib.h, **333**
include/Parser/Parser_AG.h, **334**
include/Parser/Semantics_actions.h, **335**
include/Parser/Semantics_checks.h, **337**
include/Util/Utilities.h, **338**
increment_index_syntax_order
 genevalmag::Semantics_checks, **282**
increment_level
 genevalmag, **36**
increment_precedence_level
 genevalmag::Semantics_checks, **283**
index_access_eq
 genevalmag, **53**
index_ocurrence
 genevalmag::k_p_project, **250**
index_syntax_order
 genevalmag::Semantics_checks, **283**
inic_func
 genevalmag, **37**
init_order_ag
 genevalmag::Builder_plans, **154**
inorder_only_instance
 genevalmag::Equation, **174**
inorder_only_leaf
 genevalmag::Equation, **174**
ins_attr_computed
 genevalmag, **37**
is_comparable
 genevalmag::Expr_function, **185**
is_infix
 genevalmag::Expr_function, **185**
is_inherit
 genevalmag::Attribute, **86**
is_non_terminal
 genevalmag::Symbol, **302**
IS_OPERATOR
 genevalmag, **37**
is_operator
 genevalmag::Function, **237**
is_postfix
 genevalmag::Expr_function, **186**
is_prefix
 genevalmag::Expr_function, **186**
is_synthetize
 genevalmag::Attribute, **86**
is_type_basic
 genevalmag::Sort, **291**
item
 genevalmag::i_w, **246**
Item_work
 genevalmag, **15**
iterator_f
 genevalmag, **15**
iterator_t
 genevalmag, **15**

k_bool
 genevalmag, **16**
k_char
 genevalmag, **16**
k_float
 genevalmag, **16**
k_infix
 genevalmag, **17**
k_inherit
 genevalmag, **17**
k_int
 genevalmag, **16**
k_left
 genevalmag, **16**
k_non_assoc
 genevalmag, **16**
k_non_terminal
 genevalmag, **17**
k_postfix
 genevalmag, **17**
k_prefix
 genevalmag, **17**
k_right
 genevalmag, **16**
k_string
 genevalmag, **16**
k_synthetize
 genevalmag, **17**
k_terminal
 genevalmag, **17**
key
 genevalmag::Attribute, **86**
 genevalmag::Equation, **174**
 genevalmag::Expr_instance, **198**
 genevalmag::Function, **237**
 genevalmag::Rule, **271**
 genevalmag::Sort, **292**
 genevalmag::Symbol, **302**
Key_plan
 genevalmag, **15**
Key_plan_project
 genevalmag, **15**

Key_work_list
 genevalmag, 15
 kind
 genevalmag::vertex_data_t, 306
 l_value
 genevalmag::Equation, 177
 LEAVE
 genevalmag, 38
 literal_type
 genevalmag, 16
 load_attributes
 genevalmag::Attr_grammar, 76

 maggen.cpp
 main, 360
 Maglib
 genevalmag::Maglib, 255
 main
 maggen.cpp, 360
 merge_graph
 utilities, 58
 merge_vec
 genevalmag, 38
 merge_vec_without_plan
 genevalmag, 38

 newAttrs
 genevalmag, 53

 oper_assoc
 genevalmag, 16
 oper_mode
 genevalmag, 16
 operator<
 genevalmag::c_rule, 164
 genevalmag::k_p_project, 249
 genevalmag::k_plan, 251
 operator<<
 genevalmag, 39
 operator=
 genevalmag::Attribute, 87
 genevalmag::c_rule, 164
 genevalmag::Equation, 175
 genevalmag::Expr_function, 187
 genevalmag::Expr_instance, 198
 genevalmag::Expr_literal, 212
 genevalmag::Function, 238
 genevalmag::i_w, 246
 genevalmag::k_p_project, 249
 genevalmag::k_plan, 251
 genevalmag::k_w, 253
 genevalmag::Rule, 271
 genevalmag::Sort, 292

 genevalmag::Symbol, 303
 operator==
 genevalmag::i_w, 246
 genevalmag::k_p_project, 249
 genevalmag::k_plan, 252
 genevalmag::k_w, 253
 order_attr
 genevalmag::i_w, 246
 Order_eval_eq
 genevalmag, 15
 Order_rule
 genevalmag, 16

 p_Adp_graphs
 genevalmag::Builder_graphs, 136
 p_Adp_subgraphs_cyclics
 genevalmag::Builder_graphs, 136
 p_Dcg_graphs
 genevalmag::Builder_graphs, 137
 p_Down_graphs
 genevalmag::Builder_graphs, 137
 p_Dp_graphs
 genevalmag::Builder_graphs, 137
 parent
 genevalmag::Expression, 228
 parse_grammar
 genevalmag::Parser_AG, 258
 parse_parameters
 genevalmag, 39
 Parser_AG
 genevalmag::Parser_AG, 258
 PATH_OUT_PLAN
 genevalmag, 39
 PATH_OUT_PLAN_PROJECT
 genevalmag, 40
 path_output
 genevalmag::Builder_code, 121
 PATH_OUTPUT_ADP
 genevalmag, 40
 PATH_OUTPUT_CYCLIC
 genevalmag, 40
 PATH_OUTPUT_DCG
 genevalmag, 40
 PATH_OUTPUT_DOWN
 genevalmag, 40
 PATH_OUTPUT_DP
 genevalmag, 41
 PATH_OUTPUT_GRAPHS
 genevalmag, 41
 plan
 genevalmag::k_plan, 252
 plan_family_computed
 genevalmag, 41

plans_project
 genevalmag::Builder_plans, 154

plans_project_uniques
 genevalmag::Builder_plans, 154

plans_uniques
 genevalmag::Builder_plans, 155

precedence_level
 genevalmag::Expr_function, 190
 genevalmag::Semantics_checks, 283

print_all_visit_sequences
 genevalmag::Builder_visit_sequences,
 162

print_graph
 utilities, 59

project_graph
 utilities, 59

purge_plan_with
 genevalmag, 42

push_mark
 genevalmag, 42

r_att_grammar
 genevalmag::attritute_-
 grammar::definition, 97

r_attribute_st
 genevalmag::attritute_-
 grammar::definition, 97

r_attributes
 genevalmag::attritute_-
 grammar::definition, 97

r_bloq_sem
 genevalmag::attritute_-
 grammar::definition, 97

r_boolean
 genevalmag::attritute_-
 grammar::definition, 97

r_char
 genevalmag::attritute_-
 grammar::definition, 97

r_compute_eq
 genevalmag::attritute_-
 grammar::definition, 98

r_conj_symb
 genevalmag::attritute_-
 grammar::definition, 98

r_cpp_basic_types
 genevalmag::attritute_-
 grammar::definition, 98

r_cpp_reserved_words
 genevalmag::attritute_-
 grammar::definition, 98

r_decl_attr
 genevalmag::attritute_-
 grammar::definition, 98

r_decl_func
 genevalmag::attritute_-
 grammar::definition, 98

r_decl_oper
 genevalmag::attritute_-
 grammar::definition, 98

r_decl_rule
 genevalmag::attritute_-
 grammar::definition, 98

r_decl_sort
 genevalmag::attritute_-
 grammar::definition, 99

r_dom_func
 genevalmag::attritute_-
 grammar::definition, 99

r_eqs
 genevalmag::Rule, 274

r_equation
 genevalmag::attritute_-
 grammar::definition, 99

r_esc_seq
 genevalmag::attritute_-
 grammar::definition, 99

r_expr_prime
 genevalmag::attritute_-
 grammar::definition, 99

r_expr_prime_prime
 genevalmag::attritute_-
 grammar::definition, 99

r_expression
 genevalmag::attritute_-
 grammar::definition, 99

r_function
 genevalmag::attritute_-
 grammar::definition, 100

r_function_st
 genevalmag::attritute_-
 grammar::definition, 100

r_id
 genevalmag::Rule, 275

r_id_op
 genevalmag::attritute_-
 grammar::definition, 100

r_ident
 genevalmag::attritute_-
 grammar::definition, 100

r_instance
 genevalmag::attritute_-
 grammar::definition, 100

r_left_symbol
 genevalmag::Rule, 275

r_literal
 genevalmag::attritute_-

grammar::definition, 100
r_non_term_st
 genevalmag::attribute_-
 grammar::definition, 100
r_offset
 genevalmag::Rule, 275
r_op_infix_st
 genevalmag::attribute_-
 grammar::definition, 101
r_op_postfix_st
 genevalmag::attribute_-
 grammar::definition, 101
r_op_prefix_st
 genevalmag::attribute_-
 grammar::definition, 101
r_oper
 genevalmag::attribute_-
 grammar::definition, 101
r_oper_assoc
 genevalmag::attribute_-
 grammar::definition, 101
r_oper_infix
 genevalmag::attribute_-
 grammar::definition, 101
r_oper_mode
 genevalmag::attribute_-
 grammar::definition, 101
r_oper_postfix
 genevalmag::attribute_-
 grammar::definition, 102
r_oper_prefix
 genevalmag::attribute_-
 grammar::definition, 102
r_reserved_word
 genevalmag::attribute_-
 grammar::definition, 102
r_right_rule
 genevalmag::attribute_-
 grammar::definition, 102
r_right_side
 genevalmag::Rule, 275
r_rules
 genevalmag::attribute_-
 grammar::definition, 102
r_semantic_domain
 genevalmag::attribute_-
 grammar::definition, 102
r_sort_st
 genevalmag::attribute_-
 grammar::definition, 102
r_sort_stable
 genevalmag::attribute_-
 grammar::definition, 103

r_string
 genevalmag::attribute_-
 grammar::definition, 103
r_string_lit
 genevalmag::attribute_-
 grammar::definition, 103
r_terminal
 genevalmag::attribute_-
 grammar::definition, 103
r_type_attr
 genevalmag::attribute_-
 grammar::definition, 103
r_value
 genevalmag::Equation, 177
replace_child
 genevalmag::Expr_node, 219
reset_semantic_context
 genevalmag::Semantics_checks, 283
return_index_context
 genevalmag::Builder_plans, 148
return_index_plan
 genevalmag::Builder_plans, 149
return_index_plan_p
 genevalmag::Builder_plans, 149
return_index_vec
 genevalmag, 42
return_sort
 genevalmag::Attr_grammar, 76
return_vertex
 utilities, 60
Rule
 genevalmag::Rule, 262
rule_exp
 genevalmag::attribute_-
 grammar::definition, 93
rule_lexeme
 genevalmag::attribute_-
 grammar::definition, 93

s_ins
 genevalmag::Sort, 294
s_name
 genevalmag::Sort, 294
s_type_basic
 genevalmag::Sort, 294
save_adp_graphs
 genevalmag::Builder_graphs, 130
save_all_graphs
 genevalmag::Builder_graphs, 131
 genevalmag::Builder_plans, 150
save_all_plans
 genevalmag::Builder_plans, 151
save_all_plans_project

genevalmag::Builder_plans, 152
save_assoc_op
 genevalmag, 43
save_attr_ins
 genevalmag, 43
save_cyclic_graphs
 genevalmag::Builder_graphs, 132
 genevalmag::Builder_plans, 153
save_dcg_graphs
 genevalmag::Builder_graphs, 133
save_domain_func
 genevalmag, 44
save_down_graphs
 genevalmag::Builder_graphs, 134
save_dp_graphs
 genevalmag::Builder_graphs, 135
save_grammar_file
 genevalmag::Parser_AG, 259
save_image_func
 genevalmag, 44
save_index_ins
 genevalmag, 45
save_member_list_attr
 genevalmag, 45
save_mode_op
 genevalmag, 46
save_name_func
 genevalmag, 46
save_prec_op
 genevalmag, 46
save_right_side_rule
 genevalmag, 47
save_rule
 genevalmag, 47
save_rvalue
 genevalmag, 47
save_sort_attr
 genevalmag, 48
save_type_attr
 genevalmag, 49
sem_check
 genevalmag, 53
 genevalmag::Parser_AG, 260
Semantics_checks
 genevalmag::Semantics_checks, 276
set_attr
 genevalmag::Expr_instance, 199
set_attr_grammar
 genevalmag, 49
set_function
 genevalmag::Expr_function, 188
set_id
 genevalmag::Equation, 175
genevalmag::Rule, 272
genevalmag::Symbol, 303
set_image
 genevalmag::Function, 239
set_is_operator
 genevalmag::Function, 239
set_l_value
 genevalmag::Equation, 175
set_left_symbol
 genevalmag::Rule, 272
set_member_symbol
 genevalmag::Attribute, 87
set_mod_type
 genevalmag::Attribute, 88
set_mode
 genevalmag::Function, 240
set_name
 genevalmag::Attribute, 88
 genevalmag::Function, 240
 genevalmag::Sort, 293
 genevalmag::Symbol, 303
set_num
 genevalmag::Expr_instance, 199
set_oper_assoc
 genevalmag::Function, 240
set_parent
 genevalmag::Expression, 226
set_prec
 genevalmag::Function, 241
set_precedence_level
 genevalmag::Expr_function, 188
set_r_value
 genevalmag::Equation, 176
set_sem_check
 genevalmag, 49
set_sort_type
 genevalmag::Attribute, 89
set_symb
 genevalmag::Expr_instance, 200
set_syntax_order
 genevalmag::Expr_function, 188
set_type
 genevalmag::Expr_literal, 213
 genevalmag::Symbol, 304
set_type_basic
 genevalmag::Sort, 293
set_type_synthesized
 genevalmag::Expression, 227
set_value
 genevalmag::Expr_literal, 213
show_help_information
 genevalmag, 50
signature

genevalmag::Function, 241
 skip
 genevalmag::skip_parser::definition, 286
 Sort
 genevalmag::Sort, 288, 289
 src/Attr_grammar/Attr_grammar.cpp, 340
 src/Attr_grammar/Attribute.cpp, 341
 src/Attr_grammar/Equation.cpp, 342
 src/Attr_grammar/Function.cpp, 343
 src/Attr_grammar/Rule.cpp, 344
 src/Attr_grammar/Sort.cpp, 345
 src/Attr_grammar/Symbol.cpp, 346
 src/Builders/Builder_code.cpp, 347
 src/Builders/Builder_graphs.cpp, 349
 src/Builders/Builder_plans.cpp, 351
 src/Builders/Builder_visit_sequences.cpp,
 352
 src/Expression_tree/Expr_function.cpp, 353
 src/Expression_tree/Expr_instance.cpp, 354
 src/Expression_tree/Expr_leaf.cpp, 355
 src/Expression_tree/Expr_literal.cpp, 356
 src/Expression_tree/Expr_node.cpp, 357
 src/Expression_tree/Expression.cpp, 358
 src/maggen.cpp, 359
 src/Maglib.cpp, 361
 src/Parser/Parser_AG.cpp, 362
 src/Parser/Semantics_actions.cpp, 364
 src/Parser/Semantics_checks.cpp, 367
 src/Util/Utilities.cpp, 368
 st_attributes
 genevalmag::attribute_-
 grammar::definition, 103
 st_functions
 genevalmag::attribute_-
 grammar::definition, 103
 st_non_terminal
 genevalmag::attribute_-
 grammar::definition, 104
 st_op_infix
 genevalmag::attribute_-
 grammar::definition, 104
 st_op_postfix
 genevalmag::attribute_-
 grammar::definition, 104
 st_op_prefix
 genevalmag::attribute_-
 grammar::definition, 104
 st_sorts
 genevalmag::attribute_-
 grammar::definition, 104
 stack_inner_node
 genevalmag, 53
 stack_node
 genevalmag, 53
 start
 genevalmag::attribute_-
 grammar::definition, 97
 genevalmag::skip_parser::definition, 286
 swap_root_child
 genevalmag, 50
 swap_root_grandson
 genevalmag, 50
 symb_attrs
 genevalmag::Symbol, 305
 symb_id
 genevalmag::Symbol, 305
 symb_name
 genevalmag::Symbol, 305
 symb_type
 genevalmag::Symbol, 305
 Symbol
 genevalmag::Symbol, 296
 symbol_project
 genevalmag::k_p_project, 250
 symbol_type
 genevalmag, 17
 syntax_order
 genevalmag::Expr_function, 190
 timeval_diff
 genevalmag, 51
 to_string
 genevalmag::Attr_grammar, 77
 genevalmag::Attribute, 89
 genevalmag::Equation, 176
 genevalmag::Expr_function, 189
 genevalmag::Expr_instance, 200
 genevalmag::Expr_leaf, 206
 genevalmag::Expr_literal, 214
 genevalmag::Expr_node, 220
 genevalmag::Expression, 227
 genevalmag::Function, 242
 genevalmag::Rule, 273
 genevalmag::Sort, 293
 genevalmag::Symbol, 304
 to_string_map
 genevalmag, 51
 to_string_not_eqs
 genevalmag::Rule, 273
 type
 genevalmag::Expr_literal, 214
 type_attr
 genevalmag, 17
 type_synthesized
 genevalmag::Expression, 228
 update_visit_sequence

genevalmag::Builder_visit_sequences,
 162
utilities, 54
 belong_index, 54
 clean_output_folder, 54
 cleaning_tabs, 55
 copy_static_code, 56
 create_folder, 56
 generate_names_attr, 57
 generate_names_instance, 57
 merge_graph, 58
 print_graph, 59
 project_graph, 59
 return_vertex, 60
 write_inf_context, 61

value
 genevalmag::Expr_literal, 215
Vertex
 genevalmag, 16
Visit_seq
 genevalmag, 16

warshall_algorithm
 genevalmag, 51
write_inf_context
 utilities, 61
write_vector_with_inic
 genevalmag, 51