

# SCSP CSP Solver

Generated by Doxygen 1.8.3.1

Fri Apr 19 2013 18:24:19



# Contents

<b>1</b>	<b>Class Index</b>	<b>1</b>
1.1	Class List . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	active_tuple Struct Reference . . . . .	5
3.1.1	Member Data Documentation . . . . .	5
3.1.1.1	ac_vars_from_tuple . . . . .	5
3.1.1.2	added_tuples . . . . .	5
3.1.1.3	all_ac_vars . . . . .	5
3.1.1.4	all_ac_vars_count . . . . .	5
3.1.1.5	common_cols . . . . .	5
3.1.1.6	last . . . . .	6
3.1.1.7	next . . . . .	6
3.1.1.8	pos . . . . .	6
3.1.1.9	special_generating_solutions . . . . .	6
3.1.1.10	tm . . . . .	6
3.1.1.11	tms . . . . .	6
3.1.1.12	total_added . . . . .	6
3.1.1.13	total_vars . . . . .	6
3.1.1.14	treen . . . . .	6
3.1.1.15	tuple . . . . .	6
3.1.1.16	tuple_counts . . . . .	6
3.1.1.17	var_map . . . . .	6
3.1.1.18	var_names . . . . .	6
3.1.1.19	vars . . . . .	6
3.2	btd_data Struct Reference . . . . .	6
3.2.1	Member Data Documentation . . . . .	6
3.2.1.1	goods . . . . .	6
3.2.1.2	init_vars . . . . .	6

3.2.1.3	nogoods	6
3.2.1.4	uninit_vars	6
3.3	btd_engine Struct Reference	7
3.3.1	Member Data Documentation	7
3.3.1.1	current_cluster	7
3.3.1.2	next_variable_in_current_cluster_to_select	7
3.3.1.3	tree_root	7
3.3.1.4	variable_highest_cluster	7
3.4	circular_array Struct Reference	7
3.4.1	Member Data Documentation	7
3.4.1.1	array	7
3.4.1.2	i	7
3.4.1.3	size	7
3.5	col_card Struct Reference	8
3.5.1	Member Data Documentation	8
3.5.1.1	card	8
3.5.1.2	col_name	8
3.6	comb_cdata Struct Reference	8
3.6.1	Member Data Documentation	8
3.6.1.1	cd	8
3.6.1.2	comb	8
3.6.1.3	context	8
3.7	common_var Struct Reference	8
3.7.1	Member Data Documentation	9
3.7.1.1	apos	9
3.7.1.2	bpos	9
3.8	constraint Struct Reference	9
3.8.1	Member Data Documentation	9
3.8.1.1	ac3_1_last	9
3.8.1.2	constraint_id	9
3.8.1.3	decomposition_added	9
3.8.1.4	fill	9
3.8.1.5	function	9
3.8.1.6	ht	9
3.8.1.7	lastdfid	9
3.8.1.8	name	9
3.8.1.9	parameters	9
3.8.1.10	qgn_references	9
3.8.1.11	tuples	9
3.8.1.12	uninit_vars	10

3.8.1.13	variables	10
3.9	constraint_graph Struct Reference	10
3.9.1	Member Data Documentation	10
3.9.1.1	cge_i	10
3.9.1.2	cgn_i	10
3.9.1.3	cgn_i_size	10
3.9.1.4	constraint_node_count	10
3.9.1.5	edge_count	10
3.9.1.6	edge_list	10
3.9.1.7	fill_edges_count	10
3.9.1.8	left_deep_list	10
3.9.1.9	matrix	10
3.9.1.10	max_arity	10
3.9.1.11	node_list	10
3.9.1.12	root	10
3.10	constraint_graph_edge Struct Reference	11
3.10.1	Member Data Documentation	11
3.10.1.1	common_vars	11
3.10.1.2	id	11
3.10.1.3	node1	11
3.10.1.4	node2	11
3.10.1.5	query_node	11
3.10.1.6	table	11
3.10.1.7	weight	11
3.11	constraint_graph_node Struct Reference	11
3.11.1	Member Data Documentation	12
3.11.1.1	children	12
3.11.1.2	color	12
3.11.1.3	combinations_removed	12
3.11.1.4	constr	12
3.11.1.5	fill_in	12
3.11.1.6	generation	12
3.11.1.7	hash_table	12
3.11.1.8	heap_pos	12
3.11.1.9	id	12
3.11.1.10	name	12
3.11.1.11	neighbours	12
3.11.1.12	out_of_order	12
3.11.1.13	parents	12
3.11.1.14	peo	12

3.11.1.15	pgeneration	12
3.11.1.16	table_changed_after_suspend	12
3.12	constraint_tree Struct Reference	12
3.12.1	Member Data Documentation	13
3.12.1.1	clusters	13
3.12.1.2	id	13
3.12.1.3	nodes	13
3.12.1.4	root	13
3.12.1.5	width	13
3.13	constraint_tree_node Struct Reference	13
3.13.1	Member Data Documentation	13
3.13.1.1	children	13
3.13.1.2	htables	13
3.13.1.3	id	13
3.13.1.4	parent	13
3.14	constraints Struct Reference	13
3.14.1	Member Data Documentation	14
3.14.1.1	constraint_list	14
3.14.1.2	count	14
3.14.1.3	has_intension	14
3.14.1.4	last_id	14
3.14.1.5	max_arity	14
3.14.1.6	size	14
3.14.1.7	tables	14
3.15	decision_tree Struct Reference	14
3.15.1	Member Data Documentation	14
3.15.1.1	attribute	14
3.15.1.2	id	14
3.15.1.3	left	14
3.15.1.4	level	15
3.15.1.5	parent	15
3.15.1.6	right	15
3.15.1.7	value	15
3.16	decision_tree_system Struct Reference	15
3.16.1	Member Data Documentation	15
3.16.1.1	dt	15
3.16.1.2	names	15
3.17	decomposed_tree Struct Reference	15
3.17.1	Member Data Documentation	17
3.17.1.1	adjacency	17

---

3.17.1.2	affiliate_rels	17
3.17.1.3	affiliates_map	17
3.17.1.4	ancestor_table	17
3.17.1.5	btddata	17
3.17.1.6	c_array	17
3.17.1.7	cgs	17
3.17.1.8	children	17
3.17.1.9	children_not_processed	17
3.17.1.10	clique_number	17
3.17.1.11	cluster_relation	17
3.17.1.12	cluster_var_names	17
3.17.1.13	comb_datas	17
3.17.1.14	combinations	17
3.17.1.15	combs	17
3.17.1.16	constraints	17
3.17.1.17	context_rels	17
3.17.1.18	degree	17
3.17.1.19	depth	17
3.17.1.20	dftid	17
3.17.1.21	f_tables	17
3.17.1.22	feature_stats	17
3.17.1.23	fill_edge_constraints	17
3.17.1.24	filter_cluster_relation	17
3.17.1.25	filter_cluster_relation_reverse	17
3.17.1.26	filter_more	17
3.17.1.27	glue_combs	17
3.17.1.28	has_stats	18
3.17.1.29	heap_pos	18
3.17.1.30	id	18
3.17.1.31	ignore_tables	18
3.17.1.32	in_fringe	18
3.17.1.33	leafs	18
3.17.1.34	level	18
3.17.1.35	mini_bucket_rels	18
3.17.1.36	parent	18
3.17.1.37	parent_separator_ac	18
3.17.1.38	parent_separator_tm	18
3.17.1.39	parent_separator_tm_common_cols	18
3.17.1.40	pending_sep_table_generation	18
3.17.1.41	position_in_parent_node_children_list	18

3.17.1.42 post	18
3.17.1.43 pre	18
3.17.1.44 projected_relations	18
3.17.1.45 projected_rels_on_c_vars	18
3.17.1.46 projected_rels_on_sep_vars	18
3.17.1.47 queue	18
3.17.1.48 relation_vars	18
3.17.1.49 representative_comb_datas	18
3.17.1.50 separator	18
3.17.1.51 separator_relation	18
3.17.1.52 separator_relations	18
3.17.1.53 separator_rels	18
3.17.1.54 separator_scopes	18
3.17.1.55 separator_var_names	18
3.17.1.56 separator_vars	19
3.17.1.57 separators	19
3.17.1.58 stats	19
3.17.1.59 subtree_size	19
3.17.1.60 tables	19
3.17.1.61 uncovered_vars	19
3.17.1.62 varlist	19
3.17.1.63 vars	19
3.17.1.64 visit_stamp	19
3.17.1.65 weight	19
3.18 decomposition_info Struct Reference	19
3.18.1 Member Data Documentation	19
3.18.1.1 hyperwidth	19
3.18.1.2 largest_separator_size	19
3.18.1.3 largest_separator_size_rels	19
3.18.1.4 mhyperwidth	19
3.18.1.5 min_hyperwidth	19
3.18.1.6 mtreewidth	20
3.18.1.7 number_of_clusters	20
3.18.1.8 smallest_separator_size	20
3.18.1.9 smallest_separator_size_rels	20
3.18.1.10 treewidth	20
3.19 gbtd_system Struct Reference	20
3.19.1 Member Data Documentation	20
3.19.1.1 ac	20
3.19.1.2 allSol	20



3.19.1.3	cd	20
3.19.1.4	count_on_m_s	20
3.19.1.5	current_cluster	21
3.19.1.6	current_var	21
3.19.1.7	current_var_p	21
3.19.1.8	isolCount	21
3.19.1.9	queue	21
3.19.1.10	queues	21
3.19.1.11	revision_time	21
3.19.1.12	root	21
3.19.1.13	solCount	21
3.19.1.14	system	21
3.19.1.15	total_clusters	21
3.19.1.16	total_vars	21
3.19.1.17	treens	21
3.19.1.18	tuple_position_htid	21
3.19.1.19	tuple_positions	21
3.19.1.20	unassigned_vars	21
3.19.1.21	vars	21
3.19.1.22	wtn	21
3.20	hashmap Struct Reference	21
3.20.1	Member Data Documentation	22
3.20.1.1	col_pos	22
3.20.1.2	datap	22
3.20.1.3	hashmap	22
3.20.1.4	ht	22
3.20.1.5	id	22
3.20.1.6	inserted	22
3.20.1.7	iterator	22
3.20.1.8	iteratorptr	22
3.20.1.9	longest_chain	22
3.20.1.10	max	22
3.20.1.11	note	22
3.20.1.12	single	22
3.20.1.13	size	22
3.20.1.14	width	22
3.21	heap Struct Reference	22
3.21.1	Member Data Documentation	23
3.21.1.1	body	23
3.21.1.2	count	23

3.21.1.3	size	23
3.22	histogram Struct Reference	23
3.22.1	Member Data Documentation	23
3.22.1.1	columns	23
3.22.1.2	dist	23
3.22.1.3	histo	23
3.22.1.4	max	23
3.22.1.5	min	23
3.22.1.6	values	23
3.23	htable Struct Reference	23
3.23.1	Member Data Documentation	25
3.23.1.1	assigned	25
3.23.1.2	capacity	25
3.23.1.3	cgn	25
3.23.1.4	clustered1	25
3.23.1.5	clustered2	25
3.23.1.6	col_names	25
3.23.1.7	col_names_l	25
3.23.1.8	com_cols_with_next_in_order	25
3.23.1.9	comb_datas	25
3.23.1.10	complete	25
3.23.1.11	constr	25
3.23.1.12	context	25
3.23.1.13	deleted	25
3.23.1.14	deleted_time_node	25
3.23.1.15	deleted_times	25
3.23.1.16	entropy	25
3.23.1.17	flag	25
3.23.1.18	histo	25
3.23.1.19	htiterator	25
3.23.1.20	id	25
3.23.1.21	in_combinations	25
3.23.1.22	in_combinations_hashed	25
3.23.1.23	in_tree	25
3.23.1.24	incomplete_clustereda	25
3.23.1.25	incomplete_clusteredb	25
3.23.1.26	incomplete_i	25
3.23.1.27	incomplete_j	25
3.23.1.28	index_of_ht_in_combinations	26
3.23.1.29	is_first	26

3.23.1.30	iterator	26
3.23.1.31	join_cutoff	26
3.23.1.32	jcolpos	26
3.23.1.33	jthm	26
3.23.1.34	jthmht	26
3.23.1.35	last	26
3.23.1.36	last_pos	26
3.23.1.37	markgeneration	26
3.23.1.38	name	26
3.23.1.39	neighbour_list_per_tuple	26
3.23.1.40	next	26
3.23.1.41	no_destroy	26
3.23.1.42	prev	26
3.23.1.43	problemtable	26
3.23.1.44	short_name	26
3.23.1.45	supports	26
3.23.1.46	table	26
3.23.1.47	tmpid	26
3.23.1.48	tn	26
3.23.1.49	total_high	26
3.23.1.50	tree_maps	26
3.23.1.51	tuple_list	26
3.23.1.52	tuple_list_array	26
3.23.1.53	tuple_lost	26
3.23.1.54	tuple_pointed_at_from	26
3.23.1.55	tuple_values	26
3.23.1.56	type	27
3.23.1.57	vars	27
3.23.1.58	width	27
3.24	htv_undo_data Struct Reference	27
3.24.1	Member Data Documentation	27
3.24.1.1	htv	27
3.24.1.2	list_of_htvar_toundo_domlist_stack	27
3.24.1.3	min_marks	27
3.24.1.4	undo_deg	27
3.25	htvar Struct Reference	27
3.25.1	Member Data Documentation	28
3.25.1.1	cd_cursor	28
3.25.1.2	chosen_value	28
3.25.1.3	current_domain_tm_domlist	28

3.25.1.4	dom_count	28
3.25.1.5	domlist_stack	28
3.25.1.6	ht	28
3.25.1.7	tm_cursor	28
3.25.1.8	value_lock	28
3.25.1.9	vid	28
3.26	lastGAC Struct Reference	28
3.26.1	Member Data Documentation	28
3.26.1.1	ht	28
3.26.1.2	pos	28
3.26.1.3	vpos	28
3.27	light_stack Struct Reference	28
3.27.1	Member Data Documentation	29
3.27.1.1	head	29
3.28	light_stack_node Struct Reference	29
3.28.1	Member Data Documentation	29
3.28.1.1	body	29
3.28.1.2	next	29
3.28.1.3	value	29
3.29	llist Struct Reference	29
3.29.1	Member Data Documentation	30
3.29.1.1	count	30
3.29.1.2	head	30
3.29.1.3	id	30
3.29.1.4	max	30
3.29.1.5	min	30
3.29.1.6	tail	30
3.30	llist_node Struct Reference	30
3.30.1	Member Data Documentation	30
3.30.1.1	body	30
3.30.1.2	ibody	30
3.30.1.3	list	30
3.30.1.4	next	30
3.30.1.5	previous	30
3.31	main_structure Struct Reference	30
3.31.1	Member Data Documentation	33
3.31.1.1	ac3	33
3.31.1.2	active_htables	33
3.31.1.3	alive_t	33
3.31.1.4	all_solutions	33

3.31.1.5	all_tree_hts	33
3.31.1.6	apply_td	33
3.31.1.7	at	33
3.31.1.8	auto_all_ht_revise_count	33
3.31.1.9	auto_all_ht_revise_count_k	33
3.31.1.10	auto_all_revise_count	33
3.31.1.11	auto_one_ht_revise_count	33
3.31.1.12	auto_one_ht_revise_count_k	33
3.31.1.13	auto_one_revise_count	33
3.31.1.14	bayesian	33
3.31.1.15	bt_counter	33
3.31.1.16	bt_visit_counter	33
3.31.1.17	cg	33
3.31.1.18	check_counter	33
3.31.1.19	cmap	33
3.31.1.20	comb_ids	33
3.31.1.21	comb_size	34
3.31.1.22	comb_size_search	34
3.31.1.23	cons	34
3.31.1.24	consistant_print	34
3.31.1.25	consistant_solution_count	34
3.31.1.26	constraint_placement	34
3.31.1.27	construct_solution	34
3.31.1.28	db_algo	34
3.31.1.29	dcid	34
3.31.1.30	delme	34
3.31.1.31	di	34
3.31.1.32	domains	34
3.31.1.33	dts	34
3.31.1.34	effectuated_vars_by_gac	34
3.31.1.35	expression_pointer	34
3.31.1.36	expressions	34
3.31.1.37	fc_cbj_fc	34
3.31.1.38	feature_vector_options	34
3.31.1.39	filename	34
3.31.1.40	forward_check	34
3.31.1.41	function_offset	34
3.31.1.42	h	34
3.31.1.43	ht_id	34
3.31.1.44	init_count	34

3.31.1.45 just_combinations . . . . .	34
3.31.1.46 just_redundancy_removal . . . . .	34
3.31.1.47 left_deep . . . . .	34
3.31.1.48 log . . . . .	34
3.31.1.49 long_tuple_index . . . . .	35
3.31.1.50 m . . . . .	35
3.31.1.51 max_cluster_tables . . . . .	35
3.31.1.52 max_domain_value . . . . .	35
3.31.1.53 max_gen_rels . . . . .	35
3.31.1.54 max_gen_scope . . . . .	35
3.31.1.55 max_gen_tuples . . . . .	35
3.31.1.56 minibucket . . . . .	35
3.31.1.57 minibucketopts . . . . .	35
3.31.1.58 next . . . . .	35
3.31.1.59 no_check_insol_tuples . . . . .	35
3.31.1.60 no_support . . . . .	35
3.31.1.61 no_tuples . . . . .	35
3.31.1.62 node_merge . . . . .	35
3.31.1.63 nosingleedge . . . . .	35
3.31.1.64 number_of_solutions . . . . .	35
3.31.1.65 only_check_connected . . . . .	35
3.31.1.66 partition_id . . . . .	35
3.31.1.67 pprs . . . . .	35
3.31.1.68 previous . . . . .	35
3.31.1.69 print_chk . . . . .	35
3.31.1.70 problemname . . . . .	35
3.31.1.71 profile . . . . .	35
3.31.1.72 profile_times . . . . .	35
3.31.1.73 project . . . . .	35
3.31.1.74 queue . . . . .	35
3.31.1.75 r . . . . .	35
3.31.1.76 r_star_algo . . . . .	35
3.31.1.77 r_star_all_single_sol . . . . .	36
3.31.1.78 r_star_gac . . . . .	36
3.31.1.79 rel_revise_counter . . . . .	36
3.31.1.80 relations . . . . .	36
3.31.1.81 remaining_nodes . . . . .	36
3.31.1.82 remove_redundant_edges . . . . .	36
3.31.1.83 rpwc . . . . .	36
3.31.1.84 rs2gbtd . . . . .	36

3.31.1.85 rs2tree_scheme . . . . .	36
3.31.1.86 rs2tree_scheme_options . . . . .	36
3.31.1.87 rs2tree_scheme_options_parameter . . . . .	36
3.31.1.88 search . . . . .	36
3.31.1.89 sol . . . . .	36
3.31.1.90 solve . . . . .	36
3.31.1.91 split_cutoff . . . . .	36
3.31.1.92 split_time . . . . .	36
3.31.1.93 startt . . . . .	36
3.31.1.94 temp . . . . .	36
3.31.1.95 temp_count . . . . .	36
3.31.1.96 temp_int_p . . . . .	36
3.31.1.97 tempp . . . . .	36
3.31.1.98 tempstr . . . . .	36
3.31.1.99 tms_list . . . . .	36
3.31.1.100tms_total_inserted . . . . .	36
3.31.1.101to_queue_supports . . . . .	36
3.31.1.102total_nodes . . . . .	36
3.31.1.103trace_trigger . . . . .	36
3.31.1.104tree_bt_time . . . . .	36
3.31.1.105tree_decom . . . . .	37
3.31.1.106tree_decom_solve . . . . .	37
3.31.1.107tree_decom_solve_1_all . . . . .	37
3.31.1.108tree_decom_wb . . . . .	37
3.31.1.109updated_combinations . . . . .	37
3.31.1.110use_histo . . . . .	37
3.31.1.111usedhts_list . . . . .	37
3.31.1.112v . . . . .	37
3.31.1.113valid . . . . .	37
3.31.1.114variable_choice_huristic . . . . .	37
3.31.1.115vars . . . . .	37
3.31.1.116visit_counter . . . . .	37
3.31.1.117vmap . . . . .	37
3.31.1.118vmap_dirty_list . . . . .	37
3.31.1.119with_index_delete_updating . . . . .	37
3.31.1.120with_tuple_support_recording . . . . .	37
3.32 node_graph Struct Reference . . . . .	37
3.32.1 Member Data Documentation . . . . .	38
3.32.1.1 adjacency_list . . . . .	38
3.32.1.2 fill_in . . . . .	38

3.32.1.3	heap_pos	38
3.32.1.4	marked_neighbours	38
3.32.1.5	matrix	38
3.32.1.6	node_count	38
3.32.1.7	node_map	38
3.32.1.8	out_of_order	38
3.32.1.9	peo	38
3.32.1.10	vid	38
3.33	parameter Struct Reference	38
3.33.1	Member Data Documentation	38
3.33.1.1	value	38
3.33.1.2	value_type	38
3.34	pseudo_cluster Struct Reference	38
3.34.1	Member Data Documentation	39
3.34.1.1	all_sol_state	39
3.34.1.2	assigned_vars	39
3.34.1.3	children	39
3.34.1.4	cluster_id	39
3.34.1.5	dfs_state	39
3.34.1.6	goods	39
3.34.1.7	id	39
3.34.1.8	isolCount	39
3.34.1.9	nogoods	39
3.34.1.10	parent	39
3.34.1.11	parent_list_self	39
3.34.1.12	sep_commons	39
3.34.1.13	separator_vars	39
3.34.1.14	solCount	39
3.34.1.15	treen	39
3.34.1.16	unassigned_count	39
3.34.1.17	unassigned_vars	40
3.34.1.18	undo_info	40
3.35	rb_tree Struct Reference	40
3.35.1	Member Data Documentation	40
3.35.1.1	color	40
3.35.1.2	id	40
3.35.1.3	ikey	40
3.35.1.4	key	40
3.35.1.5	left	40
3.35.1.6	parent	40



3.35.1.7	payload	40
3.35.1.8	right	40
3.36	rbt_set Struct Reference	40
3.36.1	Member Data Documentation	41
3.36.1.1	count	41
3.36.1.2	size	41
3.36.1.3	tmap	41
3.37	rs2tree_configuration Struct Reference	41
3.37.1	Member Data Documentation	41
3.37.1.1	queue	41
3.37.1.2	queues	41
3.37.1.3	queues_count	41
3.37.1.4	treens	41
3.37.1.5	usedhts_list	41
3.38	rs2tree_stats Struct Reference	41
3.38.1	Member Data Documentation	42
3.38.1.1	all_relations	42
3.38.1.2	avg_tuple_per_vvp	42
3.38.1.3	avg_tuple_per_vvp_log	42
3.38.1.4	avg_tuple_per_vvp_log_p	42
3.38.1.5	avg_tuple_per_vvp_norm	43
3.38.1.6	avg_tuple_per_vvp_norm_p	43
3.38.1.7	avg_tuple_per_vvp_p	43
3.38.1.8	avg_var_per_rel	43
3.38.1.9	constraints	43
3.38.1.10	deleted0	43
3.38.1.11	deleted1	43
3.38.1.12	domain_product0	43
3.38.1.13	entropy_prod_avg	43
3.38.1.14	entropy_prod_std	43
3.38.1.15	entropy_sum	43
3.38.1.16	f_factor	43
3.38.1.17	id	43
3.38.1.18	kappa	43
3.38.1.19	log_domain_sum	43
3.38.1.20	log_tuples_sum	43
3.38.1.21	processed	43
3.38.1.22	relations	43
3.38.1.23	sol_count	43
3.38.1.24	sorted_overlaps	43

3.38.1.25 sorted_overlaps_keys	43
3.38.1.26 std_tuple_per_vvp	43
3.38.1.27 std_tuple_per_vvp_log	43
3.38.1.28 std_tuple_per_vvp_log_p	43
3.38.1.29 std_tuple_per_vvp_norm	43
3.38.1.30 std_tuple_per_vvp_norm_p	43
3.38.1.31 std_tuple_per_vvp_p	43
3.38.1.32 std_var_per_rel	43
3.38.1.33 sum0	44
3.38.1.34 sum1	44
3.38.1.35 time	44
3.38.1.36 tn	44
3.38.1.37 tuple_vvp_avg	44
3.38.1.38 tuple_vvp_avg_log	44
3.38.1.39 tuple_vvp_rto	44
3.38.1.40 tuple_vvp_std	44
3.38.1.41 tuple_vvp_std_log	44
3.38.1.42 varlist	44
3.39 s_node Struct Reference	44
3.39.1 Member Data Documentation	44
3.39.1.1 body	44
3.39.1.2 key	44
3.40 separator_info Struct Reference	44
3.40.1 Member Data Documentation	45
3.40.1.1 rel_count	45
3.40.1.2 relation_mapped_to	45
3.40.1.3 relations	45
3.40.1.4 tables	45
3.41 set Struct Reference	45
3.41.1 Member Data Documentation	45
3.41.1.1 element_ptrs	45
3.41.1.2 last_removed	45
3.41.1.3 list	45
3.41.1.4 map	45
3.41.1.5 size	45
3.41.1.6 tmap	45
3.42 solution Struct Reference	45
3.42.1 Member Data Documentation	46
3.42.1.1 values	46
3.43 solution_structure Struct Reference	46

3.43.1	Member Data Documentation	46
3.43.1.1	size	46
3.43.1.2	solutions	46
3.44	table_map Struct Reference	46
3.44.1	Member Data Documentation	46
3.44.1.1	age	46
3.44.1.2	map	46
3.45	tree_map Struct Reference	47
3.45.1	Member Data Documentation	47
3.45.1.1	alists	47
3.45.1.2	ccount	47
3.45.1.3	compressed	47
3.45.1.4	count	47
3.45.1.5	deleted	47
3.45.1.6	http	47
3.45.1.7	id	47
3.45.1.8	indexed_cols	47
3.45.1.9	lists	47
3.46	tree_map_node Struct Reference	47
3.46.1	Member Data Documentation	48
3.46.1.1	anexts	48
3.46.1.2	atuples	48
3.46.1.3	ccount	48
3.46.1.4	full	48
3.46.1.5	id	48
3.46.1.6	lnexts	48
3.46.1.7	nexts	48
3.47	tree_maps Struct Reference	48
3.47.1	Member Data Documentation	48
3.47.1.1	ca	48
3.47.1.2	free	48
3.47.1.3	histogram	48
3.47.1.4	inserted	48
3.47.1.5	max	48
3.47.1.6	tms	48
3.47.1.7	trees	49
3.48	triplet Struct Reference	49
3.48.1	Member Data Documentation	49
3.48.1.1	ttt	49
3.49	tuple_ids Struct Reference	49

3.49.1	Member Data Documentation . . . . .	49
3.49.1.1	deletet . . . . .	49
3.49.1.2	ids . . . . .	49
3.49.1.3	tuple . . . . .	49
3.50	undo_info Struct Reference . . . . .	49
3.50.1	Member Data Documentation . . . . .	50
3.50.1.1	reduction_set . . . . .	50
3.50.1.2	reductions . . . . .	50
3.50.1.3	undo_LG . . . . .	50
3.51	undo_lastGAC Struct Reference . . . . .	50
3.51.1	Member Data Documentation . . . . .	50
3.51.1.1	to_undo . . . . .	50
3.51.1.2	vars_cons_values . . . . .	50
3.52	value Struct Reference . . . . .	50
3.52.1	Member Data Documentation . . . . .	51
3.52.1.1	domain_list_location . . . . .	51
3.52.1.2	killed . . . . .	51
3.52.1.3	name . . . . .	51
3.52.1.4	position . . . . .	51
3.52.1.5	sacremoved . . . . .	51
3.52.1.6	value . . . . .	51
3.53	var_undo_data Struct Reference . . . . .	51
3.53.1	Member Data Documentation . . . . .	51
3.53.1.1	domains . . . . .	51
3.53.1.2	relations . . . . .	51
3.53.1.3	tuples . . . . .	51
3.53.1.4	var . . . . .	51
3.54	var_val Struct Reference . . . . .	51
3.54.1	Member Data Documentation . . . . .	52
3.54.1.1	value . . . . .	52
3.54.1.2	var . . . . .	52
3.55	var_val_ac Struct Reference . . . . .	52
3.55.1	Member Data Documentation . . . . .	52
3.55.1.1	valn . . . . .	52
3.55.1.2	var . . . . .	52
3.56	variable Struct Reference . . . . .	52
3.56.1	Member Data Documentation . . . . .	53
3.56.1.1	assigned_count . . . . .	53
3.56.1.2	chosen_value . . . . .	53
3.56.1.3	conflicts . . . . .	53

3.56.1.4	<a href="#">conflicts_to_remove</a>	53
3.56.1.5	<a href="#">constraints</a>	53
3.56.1.6	<a href="#">constraints_rb_tree</a>	53
3.56.1.7	<a href="#">current_domain</a>	53
3.56.1.8	<a href="#">current_domain_hashmap</a>	53
3.56.1.9	<a href="#">current_domain_hidden</a>	53
3.56.1.10	<a href="#">current_domain_set</a>	53
3.56.1.11	<a href="#">domain</a>	53
3.56.1.12	<a href="#">domain_size</a>	53
3.56.1.13	<a href="#">future</a>	53
3.56.1.14	<a href="#">htables</a>	53
3.56.1.15	<a href="#">htreductions</a>	53
3.56.1.16	<a href="#">init_order</a>	53
3.56.1.17	<a href="#">last_domain_update_time</a>	53
3.56.1.18	<a href="#">last_gac</a>	53
3.56.1.19	<a href="#">max_domain_value</a>	53
3.56.1.20	<a href="#">name</a>	53
3.56.1.21	<a href="#">neighbours</a>	54
3.56.1.22	<a href="#">occurance_count</a>	54
3.56.1.23	<a href="#">orig_constraint_count</a>	54
3.56.1.24	<a href="#">past</a>	54
3.56.1.25	<a href="#">reductions</a>	54
3.56.1.26	<a href="#">removed</a>	54
3.56.1.27	<a href="#">static_order_position</a>	54
3.56.1.28	<a href="#">time</a>	54
3.56.1.29	<a href="#">ui</a>	54
3.56.1.30	<a href="#">variable_number</a>	54
3.57	<a href="#">variables Struct Reference</a>	54
3.57.1	<a href="#">Member Data Documentation</a>	54
3.57.1.1	<a href="#">assigned_count</a>	54
3.57.1.2	<a href="#">count</a>	54
3.57.1.3	<a href="#">map_to_id</a>	54
3.57.1.4	<a href="#">max_domain</a>	54
3.57.1.5	<a href="#">min_domain</a>	54
3.57.1.6	<a href="#">size</a>	54
3.57.1.7	<a href="#">variable_list</a>	55
3.57.1.8	<a href="#">variable_list_init</a>	55
<b>4</b>	<b><a href="#">File Documentation</a></b>	<b>57</b>
4.1	<a href="#">ac.c File Reference</a>	57

4.1.1	Typedef Documentation . . . . .	57
4.1.1.1	var_val_ac . . . . .	57
4.1.2	Function Documentation . . . . .	57
4.1.2.1	ac3 . . . . .	57
4.1.2.2	ac4 . . . . .	57
4.1.2.3	check_arc_consist . . . . .	57
4.2	ac.h File Reference . . . . .	58
4.2.1	Detailed Description . . . . .	58
4.2.2	Function Documentation . . . . .	58
4.2.2.1	ac3 . . . . .	58
4.2.2.2	ac4 . . . . .	58
4.2.2.3	check_arc_consist . . . . .	58
4.3	active_tuple.c File Reference . . . . .	58
4.3.1	Function Documentation . . . . .	59
4.3.1.1	add_active_tuple . . . . .	59
4.3.1.2	destroy_active_tuple . . . . .	59
4.3.1.3	destroy_tree_maps . . . . .	59
4.3.1.4	extend_tuple . . . . .	59
4.3.1.5	extend_tuple_with_single_value . . . . .	59
4.3.1.6	get_stored_weight_for_tuple . . . . .	59
4.3.1.7	get_stored_weight_for_tuple_from_tree . . . . .	59
4.3.1.8	new_active_tuple . . . . .	59
4.3.1.9	new_active_tuple_sizeable . . . . .	59
4.3.1.10	new_tree_maps . . . . .	59
4.3.1.11	pop_tuple . . . . .	59
4.3.1.12	reset_ac_tm . . . . .	59
4.3.1.13	store_tuple . . . . .	59
4.3.1.14	store_tuple_payload . . . . .	59
4.3.1.15	store_weight_for_tuple . . . . .	59
4.4	active_tuple.h File Reference . . . . .	59
4.4.1	Detailed Description . . . . .	60
4.4.2	Typedef Documentation . . . . .	60
4.4.2.1	active_tuple . . . . .	60
4.4.2.2	tree_maps . . . . .	60
4.4.3	Function Documentation . . . . .	60
4.4.3.1	add_active_tuple . . . . .	60
4.4.3.2	destroy_active_tuple . . . . .	60
4.4.3.3	destroy_tree_maps . . . . .	60
4.4.3.4	extend_tuple . . . . .	60
4.4.3.5	extend_tuple_with_single_value . . . . .	60

4.4.3.6	get_stored_weight_for_tuple	61
4.4.3.7	get_stored_weight_for_tuple_from_tree	61
4.4.3.8	new_active_tuple	61
4.4.3.9	new_active_tuple_sizeable	61
4.4.3.10	new_tree_maps	61
4.4.3.11	pop_tuple	61
4.4.3.12	reset_ac_tm	61
4.4.3.13	set_tuple_int_weight	61
4.4.3.14	store_tuple	61
4.4.3.15	store_tuple_payload	61
4.4.3.16	store_weight_for_tuple	61
4.5	btdallsol.c File Reference	61
4.5.1	Function Documentation	62
4.5.1.1	btd	62
4.5.1.2	btd2	62
4.5.1.3	btd2_rec	62
4.5.1.4	btd_assign_variable	62
4.5.1.5	btd_label	62
4.5.1.6	btd_prepare_tree_for_btd2	62
4.5.1.7	btd_reset_variable_domain	62
4.5.1.8	btd_reset_variable_domain2	62
4.5.1.9	btd_un_assign_variable	62
4.5.1.10	btd_undo_value_deletions	62
4.5.1.11	btd_unlabel	62
4.5.1.12	choose_variable_from_tree	62
4.5.1.13	find_match_good_nogood	62
4.5.1.14	find_match_good_nogood_from_tree	62
4.5.1.15	init_btd_engine	62
4.5.1.16	init_btd_tree	62
4.5.1.17	new_btd_data	62
4.5.1.18	new_btd_data2	62
4.5.1.19	print_separator	62
4.5.1.20	print_separator_h	62
4.5.1.21	update_good_weight	63
4.5.1.22	update_good_weight_from_tree	63
4.6	btdallsol.h File Reference	63
4.6.1	Detailed Description	63
4.6.2	Typedef Documentation	64
4.6.2.1	btd_data	64
4.6.2.2	btd_engine	64

4.6.3	Function Documentation	64
4.6.3.1	btd	64
4.6.3.2	btd2	64
4.6.3.3	btd2_rec	64
4.6.3.4	btd_assign_variable	64
4.6.3.5	btd_bcssp	64
4.6.3.6	btd_label	64
4.6.3.7	btd_prepare_tree_for_btd2	64
4.6.3.8	btd_reset_variable_domain	64
4.6.3.9	btd_reset_variable_domain2	64
4.6.3.10	btd_un_assign_variable	64
4.6.3.11	btd_undo_value_deletions	64
4.6.3.12	btd_unlabel	64
4.6.3.13	choose_variable_from_tree	64
4.6.3.14	find_match_good_nogood	64
4.6.3.15	find_match_good_nogood_from_tree	64
4.6.3.16	init_btd_engine	64
4.6.3.17	init_btd_tree	64
4.6.3.18	new_btd_data	64
4.6.3.19	new_btd_data2	64
4.6.3.20	print_separator	64
4.6.3.21	print_separator_h	64
4.6.3.22	update_good_weight	64
4.6.3.23	update_good_weight_from_tree	64
4.7	circular_array.c File Reference	65
4.7.1	Function Documentation	65
4.7.1.1	add_to_circular_array	65
4.7.1.2	destroy_circular_array	65
4.7.1.3	new_circular_array	65
4.8	circular_array.h File Reference	65
4.8.1	Detailed Description	65
4.8.2	Typedef Documentation	65
4.8.2.1	circular_array	65
4.8.3	Function Documentation	65
4.8.3.1	add_to_circular_array	65
4.8.3.2	destroy_circular_array	65
4.8.3.3	new_circular_array	66
4.9	combinations.c File Reference	66
4.9.1	Function Documentation	67
4.9.1.1	assign_next_join	67



4.9.1.2	<a href="#">backtrack_search_join</a>	67
4.9.1.3	<a href="#">backtrack_search_join_rec</a>	67
4.9.1.4	<a href="#">check_connected_subgraph</a>	67
4.9.1.5	<a href="#">check_constraint</a>	67
4.9.1.6	<a href="#">choose_var_join</a>	67
4.9.1.7	<a href="#">clear_cmarks</a>	67
4.9.1.8	<a href="#">collect_solutions</a>	67
4.9.1.9	<a href="#">collect_solutions_with_all_combs</a>	67
4.9.1.10	<a href="#">comb</a>	67
4.9.1.11	<a href="#">combine_check_solution</a>	67
4.9.1.12	<a href="#">count_base_k</a>	67
4.9.1.13	<a href="#">count_base_k_function</a>	67
4.9.1.14	<a href="#">count_base_kv</a>	67
4.9.1.15	<a href="#">dac_check_join</a>	67
4.9.1.16	<a href="#">dfs_on_marked</a>	67
4.9.1.17	<a href="#">forward_check_join</a>	68
4.9.1.18	<a href="#">get_bf_combinations_2</a>	68
4.9.1.19	<a href="#">get_bf_combinations_3</a>	68
4.9.1.20	<a href="#">get_bf_combinations_4</a>	68
4.9.1.21	<a href="#">get_bf_combinations_5</a>	68
4.9.1.22	<a href="#">get_bf_combinations_6</a>	68
4.9.1.23	<a href="#">get_bf_combinations_7</a>	68
4.9.1.24	<a href="#">get_bf_combinations_8</a>	68
4.9.1.25	<a href="#">get_bf_combinations_from_subgraphs</a>	68
4.9.1.26	<a href="#">get_bf_combinations_list_2</a>	68
4.9.1.27	<a href="#">get_bf_combinations_list_3</a>	68
4.9.1.28	<a href="#">get_bf_combinations_list_4</a>	68
4.9.1.29	<a href="#">get_bf_combinations_list_5</a>	68
4.9.1.30	<a href="#">get_bf_combinations_list_6</a>	68
4.9.1.31	<a href="#">get_bf_combinations_list_7</a>	68
4.9.1.32	<a href="#">get_bf_combinations_list_8</a>	68
4.9.1.33	<a href="#">get_combinations</a>	68
4.9.1.34	<a href="#">get_combinations_with</a>	68
4.9.1.35	<a href="#">get_distance_k_nodes</a>	68
4.9.1.36	<a href="#">get_distance_k_nodes_rec</a>	68
4.9.1.37	<a href="#">get_solutions_of_size</a>	68
4.9.1.38	<a href="#">graph_connected</a>	68
4.9.1.39	<a href="#">hide_get_combinations</a>	68
4.9.1.40	<a href="#">mac_check_join</a>	68
4.9.1.41	<a href="#">mark_cmap</a>	68

4.9.1.42	markup_node_tree	68
4.9.1.43	markup_node_tree_recursive	68
4.9.1.44	memo_count_base_k_function	69
4.9.1.45	multipliers	69
4.9.1.46	n_choose_i	69
4.9.1.47	print_combination	69
4.9.1.48	print_combinations	69
4.9.1.49	print_node_tree	69
4.9.1.50	undo_filtering_join	69
4.10	combinations.h File Reference	69
4.10.1	Detailed Description	70
4.10.2	Typedef Documentation	70
4.10.2.1	solution_structure	70
4.10.3	Function Documentation	70
4.10.3.1	apply_ac_join	70
4.10.3.2	assign_next_join	71
4.10.3.3	backtrack_search_join	71
4.10.3.4	backtrack_search_join_rec	71
4.10.3.5	check_connected_subgraph	71
4.10.3.6	check_constraint	71
4.10.3.7	choose_var_join	71
4.10.3.8	clear_cmarks	71
4.10.3.9	collect_solutions	71
4.10.3.10	collect_solutions_with_all_combs	71
4.10.3.11	comb	71
4.10.3.12	combine_check_solution	71
4.10.3.13	compare_nodes	71
4.10.3.14	count_base_k	71
4.10.3.15	count_base_k_function	71
4.10.3.16	count_base_kv	71
4.10.3.17	dac_check_join	71
4.10.3.18	dfs_on_marked	71
4.10.3.19	fc_join	71
4.10.3.20	forward_check_join	71
4.10.3.21	get_bf_combinations_2	71
4.10.3.22	get_bf_combinations_3	71
4.10.3.23	get_bf_combinations_4	71
4.10.3.24	get_bf_combinations_5	71
4.10.3.25	get_bf_combinations_6	71
4.10.3.26	get_bf_combinations_7	71

4.10.3.27	get_bf_combinations_8	71
4.10.3.28	get_bf_combinations_from_subgraphs	72
4.10.3.29	get_bf_combinations_list_2	72
4.10.3.30	get_bf_combinations_list_3	72
4.10.3.31	get_bf_combinations_list_4	72
4.10.3.32	get_bf_combinations_list_5	72
4.10.3.33	get_bf_combinations_list_6	72
4.10.3.34	get_bf_combinations_list_7	72
4.10.3.35	get_bf_combinations_list_8	72
4.10.3.36	get_combinations	72
4.10.3.37	get_combinations_with	72
4.10.3.38	get_distance_k_nodes	72
4.10.3.39	get_distance_k_nodes_rec	72
4.10.3.40	get_solutions_of_size	72
4.10.3.41	graph_connected	72
4.10.3.42	hide_get_combinations	72
4.10.3.43	mac_check_join	72
4.10.3.44	mark_cmap	72
4.10.3.45	markup_node_tree	72
4.10.3.46	markup_node_tree_recursive	72
4.10.3.47	memo_count_base_k_function	72
4.10.3.48	multipliers	72
4.10.3.49	n_choose_i	72
4.10.3.50	print_combination	72
4.10.3.51	print_combinations	72
4.10.3.52	print_node_tree	72
4.10.3.53	revise_combs	72
4.10.3.54	sac_check_join	72
4.10.3.55	undo_filtering_join	73
4.10.3.56	xxcount_base_k_function	73
4.11	constraints.c File Reference	73
4.11.1	Function Documentation	73
4.11.1.1	add_constraint	73
4.11.1.2	add_constraint_to_variable	73
4.11.1.3	add_htables_to_constraints	73
4.11.1.4	add_parameter_to_last_constraint	73
4.11.1.5	add_variable_to_last_constraint	74
4.11.1.6	constraints_make_this_root	74
4.11.1.7	constraints_swap_parent_child	74
4.11.1.8	dual_node_graph_from_constraints	74

4.11.1.9	<a href="#">get_constraint_tree</a>	74
4.11.1.10	<a href="#">get_dual_join_tree</a>	74
4.11.1.11	<a href="#">get_unassigned_htables</a>	74
4.11.1.12	<a href="#">init_constraints</a>	74
4.11.1.13	<a href="#">new_constraint</a>	74
4.11.1.14	<a href="#">new_constraint_tree</a>	74
4.11.1.15	<a href="#">new_constraint_tree_node</a>	74
4.11.1.16	<a href="#">print_consistant_tree</a>	74
4.11.1.17	<a href="#">print_number_of_deleted_tuples_values</a>	74
4.11.1.18	<a href="#">printClusterGraphNeato</a>	74
4.11.1.19	<a href="#">reset_ac3_1_last</a>	74
4.11.1.20	<a href="#">split_constraints_into_connected_components</a>	74
4.11.1.21	<a href="#">split_constraints_into_connected_components_list</a>	74
4.11.1.22	<a href="#">split_constraints_into_connected_components_rec</a>	74
4.11.2	Variable Documentation	74
4.11.2.1	<a href="#">constraint_ids</a>	74
4.12	constraints.h File Reference	74
4.12.1	Detailed Description	75
4.12.2	Typedef Documentation	75
4.12.2.1	<a href="#">constraint</a>	75
4.12.2.2	<a href="#">constraint_tree</a>	75
4.12.2.3	<a href="#">constraint_tree_node</a>	75
4.12.2.4	<a href="#">constraints</a>	75
4.12.2.5	<a href="#">parameter</a>	75
4.12.3	Function Documentation	75
4.12.3.1	<a href="#">add_constraint</a>	76
4.12.3.2	<a href="#">add_constraint_to_variable</a>	76
4.12.3.3	<a href="#">add_htables_to_constraints</a>	76
4.12.3.4	<a href="#">add_parameter_to_last_constraint</a>	76
4.12.3.5	<a href="#">add_variable_to_last_constraint</a>	76
4.12.3.6	<a href="#">constraints_make_this_root</a>	76
4.12.3.7	<a href="#">constraints_swap_parent_child</a>	76
4.12.3.8	<a href="#">dual_node_graph_from_constraints</a>	76
4.12.3.9	<a href="#">get_constraint_tree</a>	76
4.12.3.10	<a href="#">get_dual_join_tree</a>	76
4.12.3.11	<a href="#">get_unassigned_htables</a>	76
4.12.3.12	<a href="#">init_constraints</a>	76
4.12.3.13	<a href="#">new_constraint</a>	76
4.12.3.14	<a href="#">new_constraint_tree</a>	76
4.12.3.15	<a href="#">new_constraint_tree_node</a>	76

4.12.3.16 print_consistant_tree . . . . .	76
4.12.3.17 print_number_of_deleted_tuples_values . . . . .	76
4.12.3.18 printClusterGraphNeato . . . . .	76
4.12.3.19 reset_ac3_1_last . . . . .	76
4.12.3.20 split_constraints_into_connected_components . . . . .	76
4.12.3.21 split_constraints_into_connected_components_list . . . . .	76
4.12.3.22 split_constraints_into_connected_components_rec . . . . .	76
4.13 decomposition.c File Reference . . . . .	77
4.13.1 Function Documentation . . . . .	79
4.13.1.1 add_affiliate_rels . . . . .	79
4.13.1.2 add_decomposed_tree_child . . . . .	79
4.13.1.3 add_dual_fill_edges . . . . .	79
4.13.1.4 add_fill_edges . . . . .	79
4.13.1.5 add_fill_edges_to_node_graph . . . . .	79
4.13.1.6 add_separator_relations . . . . .	79
4.13.1.7 add_separator_relations_for_cluster . . . . .	79
4.13.1.8 append_to_comb . . . . .	79
4.13.1.9 append_to_comb_head . . . . .	79
4.13.1.10 build_decomposed_tree_vertices . . . . .	79
4.13.1.11 check_all_constraints_added . . . . .	79
4.13.1.12 clear_combdatas_from_relations . . . . .	79
4.13.1.13 clear_combdatas_from_relations_only_root . . . . .	79
4.13.1.14 cliques . . . . .	79
4.13.1.15 cluster_pre_process . . . . .	79
4.13.1.16 cluster_pre_process2 . . . . .	79
4.13.1.17 compare_combs . . . . .	79
4.13.1.18 count_dual_fill_edges . . . . .	79
4.13.1.19 count_fill_edges . . . . .	79
4.13.1.20 destroy_node_graph . . . . .	79
4.13.1.21 dual_cliques . . . . .	79
4.13.1.22 find_balancing_root . . . . .	79
4.13.1.23 find_max_degree_node . . . . .	79
4.13.1.24 find_smallest_root . . . . .	79
4.13.1.25 generate_xml . . . . .	79
4.13.1.26 get_cluster_array . . . . .	79
4.13.1.27 get_comb_node_id . . . . .	79
4.13.1.28 get_common_rels_count_from_sets . . . . .	80
4.13.1.29 get_common_vars_count_from_sets . . . . .	80
4.13.1.30 get_decomposed_tree_from_library . . . . .	80
4.13.1.31 get_decomposed_tree_from_library_rec . . . . .	80

4.13.1.32 get_dual_decomposed_tree_from_library . . . . .	80
4.13.1.33 get_dual_decomposed_tree_from_library_rec . . . . .	80
4.13.1.34 get_dual_elimination_ordering . . . . .	80
4.13.1.35 get_dual_heap_value . . . . .	80
4.13.1.36 get_elimination_ordering . . . . .	80
4.13.1.37 get_heap_value . . . . .	80
4.13.1.38 hide_comb_datas . . . . .	80
4.13.1.39 in_subtree . . . . .	80
4.13.1.40 label_depth . . . . .	80
4.13.1.41 label_levels . . . . .	80
4.13.1.42 make_combinations_for_cluster . . . . .	80
4.13.1.43 make_graph . . . . .	80
4.13.1.44 make_this_root . . . . .	80
4.13.1.45 max_cardinality . . . . .	80
4.13.1.46 merge_tree_nodes . . . . .	80
4.13.1.47 merge_tree_nodes_rec . . . . .	80
4.13.1.48 merge_with_parent . . . . .	80
4.13.1.49 new_decomposed_tree . . . . .	80
4.13.1.50 new_di . . . . .	80
4.13.1.51 new_node_graph . . . . .	80
4.13.1.52 new_node_graph_from_rels . . . . .	80
4.13.1.53 new_node_graph_wpeo . . . . .	80
4.13.1.54 ng_get_heap_value . . . . .	81
4.13.1.55 ng_set_heap_pos . . . . .	81
4.13.1.56 ng_set_heap_value . . . . .	81
4.13.1.57 normalized_projected_rels_on_c_vars . . . . .	81
4.13.1.58 print_decomposed_tree . . . . .	81
4.13.1.59 print_decomposed_tree_cluster . . . . .	81
4.13.1.60 print_decomposed_tree_cluster_dot . . . . .	81
4.13.1.61 print_decomposed_tree_node . . . . .	81
4.13.1.62 print_decomposed_tree_node_dot . . . . .	81
4.13.1.63 print_decomposed_tree_node_summary_dot . . . . .	81
4.13.1.64 print_decomposed_tree_rec . . . . .	81
4.13.1.65 print_decomposition_tree_tables . . . . .	81
4.13.1.66 print_decomposition_tree_tables_h . . . . .	81
4.13.1.67 print_filtering_report . . . . .	81
4.13.1.68 print_node_graph . . . . .	81
4.13.1.69 print_tree_report . . . . .	81
4.13.1.70 reid_decomposed_tree_nodes . . . . .	81
4.13.1.71 report_separator_quality . . . . .	81

4.13.1.72	report_tree_separators	81
4.13.1.73	report_tree_separators2	81
4.13.1.74	set_dual_heap_position	81
4.13.1.75	set_dual_heap_value	81
4.13.1.76	set_heap_position	81
4.13.1.77	set_leafs	81
4.13.1.78	set_pre_post	81
4.13.1.79	set_subtree_size	81
4.13.1.80	set_uncovered_variables	81
4.13.1.81	setup_combinations	82
4.13.1.82	split_trees	82
4.13.1.83	swap_parent_child	82
4.13.1.84	un_hide_comb_datas	82
4.14	decomposition.h File Reference	82
4.14.1	Detailed Description	84
4.14.2	Typedef Documentation	84
4.14.2.1	decomposed_tree	84
4.14.2.2	decomposition_info	84
4.14.2.3	node_graph	84
4.14.2.4	separator_info	84
4.14.3	Function Documentation	84
4.14.3.1	add_affiliate_rels	84
4.14.3.2	add_decomposed_tree_child	84
4.14.3.3	add_dual_fill_edges	84
4.14.3.4	add_fill_edges	84
4.14.3.5	add_fill_edges_to_node_graph	84
4.14.3.6	add_separator_relations	84
4.14.3.7	add_separator_relations_for_cluster	84
4.14.3.8	append_to_comb	84
4.14.3.9	append_to_comb_head	84
4.14.3.10	build_decomposed_tree_vertices	85
4.14.3.11	check_all_constraints_added	85
4.14.3.12	clear_combdatas_from_relations	85
4.14.3.13	clear_combdatas_from_relations_only_root	85
4.14.3.14	cliques	85
4.14.3.15	cluster_pre_process	85
4.14.3.16	cluster_pre_process2	85
4.14.3.17	count_dual_fill_edges	85
4.14.3.18	count_fill_edges	85
4.14.3.19	destroy_node_graph	85

4.14.3.20 dual_cliques . . . . .	85
4.14.3.21 filter_tuples_for_var_val . . . . .	85
4.14.3.22 find_balancing_root . . . . .	85
4.14.3.23 find_max_degree_node . . . . .	85
4.14.3.24 find_smallest_root . . . . .	85
4.14.3.25 generate_xml . . . . .	85
4.14.3.26 get_cluster_array . . . . .	85
4.14.3.27 get_comb_node_id . . . . .	85
4.14.3.28 get_common_rels_count_from_sets . . . . .	85
4.14.3.29 get_common_vars_count_from_sets . . . . .	85
4.14.3.30 get_decomposed_tree_from_library . . . . .	85
4.14.3.31 get_decomposed_tree_from_library_rec . . . . .	85
4.14.3.32 get_dual_decomposed_tree_from_library . . . . .	85
4.14.3.33 get_dual_decomposed_tree_from_library_rec . . . . .	85
4.14.3.34 get_dual_elimination_ordering . . . . .	85
4.14.3.35 get_dual_heap_value . . . . .	85
4.14.3.36 get_elimination_ordering . . . . .	86
4.14.3.37 get_heap_value . . . . .	86
4.14.3.38 hide_comb_datas . . . . .	86
4.14.3.39 in_subtree . . . . .	86
4.14.3.40 init_cluster_queue . . . . .	86
4.14.3.41 label_depth . . . . .	86
4.14.3.42 label_levels . . . . .	86
4.14.3.43 make_combinations_for_cluster . . . . .	86
4.14.3.44 make_graph . . . . .	86
4.14.3.45 make_this_root . . . . .	86
4.14.3.46 max_cardinality . . . . .	86
4.14.3.47 merge_tree_nodes . . . . .	86
4.14.3.48 merge_tree_nodes_rec . . . . .	86
4.14.3.49 merge_with_parent . . . . .	86
4.14.3.50 new_decomposed_tree . . . . .	86
4.14.3.51 new_di . . . . .	86
4.14.3.52 new_node_graph . . . . .	86
4.14.3.53 new_node_graph_from_rels . . . . .	86
4.14.3.54 new_node_graph_wpeo . . . . .	86
4.14.3.55 ng_get_heap_value . . . . .	86
4.14.3.56 ng_set_heap_pos . . . . .	86
4.14.3.57 ng_set_heap_value . . . . .	86
4.14.3.58 normalized_projected_rels_on_c_vars . . . . .	86
4.14.3.59 print_decomposed_tree . . . . .	86



4.14.3.60	print_decomposed_tree_cluster	86
4.14.3.61	print_decomposed_tree_cluster_dot	86
4.14.3.62	print_decomposed_tree_node	86
4.14.3.63	print_decomposed_tree_node_dot	86
4.14.3.64	print_decomposed_tree_node_summary_dot	87
4.14.3.65	print_decomposed_tree_rec	87
4.14.3.66	print_decomposition_tree_tables	87
4.14.3.67	print_decomposition_tree_tables_h	87
4.14.3.68	print_filtering_report	87
4.14.3.69	print_node_graph	87
4.14.3.70	print_tree_report	87
4.14.3.71	reid_decomposed_tree_nodes	87
4.14.3.72	report_separator_quality	87
4.14.3.73	report_tree_separators	87
4.14.3.74	report_tree_separators2	87
4.14.3.75	set_dual_heap_position	87
4.14.3.76	set_dual_heap_value	87
4.14.3.77	set_heap_position	87
4.14.3.78	set_leafs	87
4.14.3.79	set_pre_post	87
4.14.3.80	set_subtree_size	87
4.14.3.81	set_uncovered_variables	87
4.14.3.82	setup_combinations	87
4.14.3.83	split_trees	87
4.14.3.84	swap_parent_child	87
4.14.3.85	un_hide_comb_datas	87
4.15	decomposition_functions.c File Reference	88
4.15.1	Function Documentation	88
4.15.1.1	add_constraints_to_cluique_pA	88
4.15.1.2	add_constraints_to_cluique_pB	88
4.15.1.3	add_constraints_to_cluique_pC	88
4.15.1.4	add_constraints_to_cluique_pD_old_not_used	88
4.15.1.5	add_constraints_to_cluique_pE	88
4.15.1.6	add_constraints_to_cluiques	88
4.15.1.7	combinations_splitter	88
4.15.1.8	generate_solutions	89
4.15.1.9	generate_solutions2	89
4.15.1.10	isolate_relations	89
4.15.1.11	process_cluster_filter_separators	89
4.15.1.12	process_cluster_generate_separator	89

4.15.1.13 set_all_relation_scope_separator . . . . .	89
4.16 decomposition_functions.h File Reference . . . . .	89
4.16.1 Detailed Description . . . . .	89
4.16.2 Function Documentation . . . . .	89
4.16.2.1 add_constraints_to_cluique_pA . . . . .	89
4.16.2.2 add_constraints_to_cluique_pB . . . . .	89
4.16.2.3 add_constraints_to_cluique_pC . . . . .	89
4.16.2.4 add_constraints_to_cluique_pD . . . . .	89
4.16.2.5 add_constraints_to_cluique_pD_old_not_used . . . . .	89
4.16.2.6 add_constraints_to_cluique_pE . . . . .	90
4.16.2.7 add_constraints_to_cluiques . . . . .	90
4.16.2.8 combinations_splitter . . . . .	90
4.16.2.9 generate_solutions . . . . .	90
4.16.2.10 generate_solutions2 . . . . .	90
4.16.2.11 isolate_relations . . . . .	90
4.16.2.12 process_cluster_filter_separators . . . . .	90
4.16.2.13 process_cluster_generate_separator . . . . .	90
4.16.2.14 set_all_relation_scope_separator . . . . .	90
4.17 do_db.c File Reference . . . . .	90
4.17.1 Function Documentation . . . . .	90
4.17.1.1 do_db . . . . .	90
4.17.1.2 do_tree . . . . .	90
4.18 do_db.h File Reference . . . . .	91
4.18.1 Detailed Description . . . . .	91
4.18.2 Function Documentation . . . . .	91
4.18.2.1 do_db . . . . .	91
4.18.2.2 do_tree . . . . .	91
4.19 domains.c File Reference . . . . .	91
4.19.1 Function Documentation . . . . .	91
4.19.1.1 add_domain_int_list_to_last_domain . . . . .	91
4.19.1.2 add_domain_name . . . . .	91
4.19.1.3 new_domains . . . . .	91
4.19.1.4 value_copy . . . . .	91
4.20 domains.h File Reference . . . . .	91
4.20.1 Detailed Description . . . . .	92
4.20.2 Typedef Documentation . . . . .	92
4.20.2.1 value . . . . .	92
4.20.3 Function Documentation . . . . .	92
4.20.3.1 add_domain_int_list_to_last_domain . . . . .	92
4.20.3.2 add_domain_name . . . . .	92

4.20.3.3	<code>new_domains</code>	92
4.20.3.4	<code>value_copy</code>	92
4.21	<code>gbtd.c</code> File Reference	92
4.21.1	Function Documentation	93
4.21.1.1	<code>advance_current_domain</code>	93
4.21.1.2	<code>assign_gvar</code>	93
4.21.1.3	<code>count_goods_nogoods</code>	93
4.21.1.4	<code>count_goods_nogoods_rec</code>	93
4.21.1.5	<code>covers_sep_relations</code>	93
4.21.1.6	<code>current_has_next</code>	93
4.21.1.7	<code>destroy_gbtd_system</code>	93
4.21.1.8	<code>destroy_htvar</code>	93
4.21.1.9	<code>destroy_pseudo_cluster</code>	93
4.21.1.10	<code>destroy_var_undo_data</code>	93
4.21.1.11	<code>do_something_with_solution</code>	93
4.21.1.12	<code>filter_gvars</code>	93
4.21.1.13	<code>find_gvar</code>	93
4.21.1.14	<code>find_gvar_wtn</code>	94
4.21.1.15	<code>generate_pseudo_tree</code>	94
4.21.1.16	<code>has_good</code>	94
4.21.1.17	<code>has_nogood</code>	94
4.21.1.18	<code>init_gbtd</code>	94
4.21.1.19	<code>init_goods_nogoods</code>	94
4.21.1.20	<code>move_to_previous_var</code>	94
4.21.1.21	<code>new_gbtd_system</code>	94
4.21.1.22	<code>new_htv_undo_data</code>	94
4.21.1.23	<code>new_htvar</code>	94
4.21.1.24	<code>new_pseudo_cluster</code>	94
4.21.1.25	<code>new_var_undo_data</code>	94
4.21.1.26	<code>no_unassigned_cover_variable</code>	94
4.21.1.27	<code>print_gbtd_system</code>	94
4.21.1.28	<code>relation_var_tuple_stamp</code>	94
4.21.1.29	<code>reset_gvar</code>	94
4.21.1.30	<code>run_gbtd</code>	94
4.21.1.31	<code>run_gbtd_wtn</code>	94
4.21.1.32	<code>set_good_nogood</code>	94
4.21.1.33	<code>sibling</code>	94
4.21.1.34	<code>unassign_subtree</code>	94
4.21.1.35	<code>undo_htv_assignment</code>	94
4.22	<code>gbtd.h</code> File Reference	94

4.22.1	Detailed Description	96
4.22.2	Typedef Documentation	96
4.22.2.1	gbtd_system	96
4.22.2.2	htv_undo_data	96
4.22.2.3	htvar	96
4.22.2.4	pseudo_cluster	96
4.22.2.5	var_undo_data	96
4.22.3	Function Documentation	96
4.22.3.1	advance_current_domain	96
4.22.3.2	assign_gvar	96
4.22.3.3	count_goods_nogoods	96
4.22.3.4	count_goods_nogoods_rec	96
4.22.3.5	covers_sep_relations	96
4.22.3.6	current_has_next	96
4.22.3.7	destroy_gbtd_system	96
4.22.3.8	destroy_htvar	96
4.22.3.9	destroy_pseudo_cluster	96
4.22.3.10	destroy_var_undo_data	96
4.22.3.11	do_something_with_solution	96
4.22.3.12	filter_gvars	96
4.22.3.13	find_gvar	96
4.22.3.14	find_gvar_wtn	96
4.22.3.15	generate_pseudo_tree	96
4.22.3.16	has_good	96
4.22.3.17	has_nogood	96
4.22.3.18	init_gbtd	97
4.22.3.19	init_goods_nogoods	97
4.22.3.20	move_to_previous_var	97
4.22.3.21	new_gbtd_system	97
4.22.3.22	new_htv_undo_data	97
4.22.3.23	new_htvar	97
4.22.3.24	new_pseudo_cluster	97
4.22.3.25	new_var_undo_data	97
4.22.3.26	no_unassigned_cover_variable	97
4.22.3.27	print_gbtd_system	97
4.22.3.28	relation_var_tuple_stamp	97
4.22.3.29	reset_gvar	97
4.22.3.30	run_gbtd	97
4.22.3.31	run_gbtd_wtn	97
4.22.3.32	set_good_nogood	97

4.22.3.33 sibling	97
4.22.3.34 unassign_subtree	97
4.22.3.35 undo_htv_assignment	97
4.23 globals.c File Reference	97
4.23.1 Function Documentation	98
4.23.1.1 add_component	98
4.23.1.2 add_single_variable_component	98
4.23.1.3 get_partition_id	98
4.23.1.4 into_components_problem	98
4.23.1.5 set_m_s_basedon_this	98
4.24 globals.h File Reference	98
4.24.1 Detailed Description	99
4.24.2 Typedef Documentation	99
4.24.2.1 main_structure	99
4.24.2.2 options	99
4.24.3 Enumeration Type Documentation	99
4.24.3.1 options	99
4.24.4 Function Documentation	101
4.24.4.1 add_component	101
4.24.4.2 add_single_variable_component	101
4.24.4.3 check_ok	101
4.24.4.4 check_ok2	101
4.24.4.5 get_partition_id	101
4.24.4.6 into_components_problem	101
4.24.4.7 set_m_s_basedon_this	101
4.25 hashmap.c File Reference	101
4.25.1 Function Documentation	101
4.25.1.1 destroy_hm	101
4.25.1.2 DoubleProbing	101
4.25.1.3 get	102
4.25.1.4 get_key	102
4.25.1.5 HashDivide	102
4.25.1.6 HashMultiply	102
4.25.1.7 hm_get_iterator	102
4.25.1.8 hm_get_next	102
4.25.1.9 insert	102
4.25.1.10 new_hashmap	102
4.25.1.11 tuple_match	102
4.26 hashmap.h File Reference	102
4.26.1 Detailed Description	102

4.26.2	Macro Definition Documentation	103
4.26.2.1	KEYTYPE	103
4.26.3	Typedef Documentation	103
4.26.3.1	hashmap	103
4.26.4	Function Documentation	103
4.26.4.1	destroy_hm	103
4.26.4.2	DoubleProbing	103
4.26.4.3	get	103
4.26.4.4	get_iterator	103
4.26.4.5	get_key	103
4.26.4.6	HashDivide	103
4.26.4.7	HashMultiply	103
4.26.4.8	hm_get_iterator	103
4.26.4.9	hm_get_next	103
4.26.4.10	insert	103
4.26.4.11	new_hashmap	103
4.26.4.12	tupple_match	103
4.27	hashtable.c File Reference	103
4.27.1	Function Documentation	106
4.27.1.1	add_name	106
4.27.1.2	add_tuple	106
4.27.1.3	add_tuple_list	106
4.27.1.4	build_histogram	106
4.27.1.5	ceil	106
4.27.1.6	clear_deleted_rows	106
4.27.1.7	cluster	106
4.27.1.8	cluster_nosort	106
4.27.1.9	cluster_sort	106
4.27.1.10	col_names_to_vars	106
4.27.1.11	compare_clustered	106
4.27.1.12	compare_col_card	106
4.27.1.13	compare_col_name	106
4.27.1.14	compare_scope	106
4.27.1.15	compare_scope_buble	106
4.27.1.16	convert_conflict_to_support2	106
4.27.1.17	convert_conflict_to_support3	106
4.27.1.18	convert_conflict_to_support4	106
4.27.1.19	convert_conflict_to_supportn	106
4.27.1.20	convert_conflict_to_supportn_rec	106
4.27.1.21	copy_col_names	106

4.27.1.22	copy_htable	106
4.27.1.23	create_fill_htable	106
4.27.1.24	delete_tuple	106
4.27.1.25	destroy_histogram	106
4.27.1.26	destroy_htable	107
4.27.1.27	destroy_htable_o	107
4.27.1.28	destroy_int_common_cols_list	107
4.27.1.29	distinct	107
4.27.1.30	do_compare	107
4.27.1.31	do_match	107
4.27.1.32	do_match_next	107
4.27.1.33	estimate_join_project_size	107
4.27.1.34	estimate_join_size	107
4.27.1.35	estimate_join_size_noproject	107
4.27.1.36	estimate_project_size	107
4.27.1.37	filter_domains_with_unary_constraints	107
4.27.1.38	filter_table	107
4.27.1.39	filter_table_with_current_domains	107
4.27.1.40	fprint_ht	107
4.27.1.41	free_tuple_ids	107
4.27.1.42	get_common_cols	107
4.27.1.43	get_domain_product	107
4.27.1.44	get_hashmap_wd	107
4.27.1.45	get_ht_iterator	107
4.27.1.46	get_int_common_col_count	107
4.27.1.47	get_int_common_cols	107
4.27.1.48	get_int_common_cols_list	107
4.27.1.49	get_int_common_cols_vars_list	107
4.27.1.50	get_int_pair	107
4.27.1.51	get_it_object	107
4.27.1.52	get_it_value	107
4.27.1.53	get_next	107
4.27.1.54	get_next_alive_index	108
4.27.1.55	get_next_tuple	108
4.27.1.56	get_projection_scope	108
4.27.1.57	get_tuple	108
4.27.1.58	get_tuple_value	108
4.27.1.59	get_value	108
4.27.1.60	get_value_sort	108
4.27.1.61	has_int_common_cols	108

4.27.1.62 hash_join_tables . . . . .	108
4.27.1.63 hash_project_distinct_table . . . . .	108
4.27.1.64 have_value . . . . .	108
4.27.1.65 init_get_next_tuple . . . . .	108
4.27.1.66 init_hashmap . . . . .	108
4.27.1.67 join_tables . . . . .	108
4.27.1.68 join_tables_new . . . . .	108
4.27.1.69 join_tables_w_suspend . . . . .	108
4.27.1.70 load_from_conflict_tuples . . . . .	108
4.27.1.71 load_from_tuples . . . . .	108
4.27.1.72 match_tuple_a_tuple_b . . . . .	108
4.27.1.73 match_tuple_single_ht . . . . .	108
4.27.1.74 merge . . . . .	108
4.27.1.75 merge_tuple . . . . .	108
4.27.1.76 merge_two_tuples_ids . . . . .	108
4.27.1.77 new_clustered_htable . . . . .	108
4.27.1.78 new_common_var . . . . .	108
4.27.1.79 new_histogram . . . . .	108
4.27.1.80 new_htable . . . . .	109
4.27.1.81 new_htable_replace . . . . .	109
4.27.1.82 new_names . . . . .	109
4.27.1.83 new_tuple_ids . . . . .	109
4.27.1.84 prepare_pipelined_join . . . . .	109
4.27.1.85 print_ht . . . . .	109
4.27.1.86 print_ht_h . . . . .	109
4.27.1.87 print_ht_line . . . . .	109
4.27.1.88 print_ht_mysql . . . . .	109
4.27.1.89 project_distinct_table . . . . .	109
4.27.1.90 project_join_tables . . . . .	109
4.27.1.91 project_table . . . . .	109
4.27.1.92 purge_htable . . . . .	109
4.27.1.93 rebuild_histogram . . . . .	109
4.27.1.94 replace_data . . . . .	109
4.27.1.95 resetset_variable_hashtables . . . . .	109
4.27.1.96 set_it_object . . . . .	109
4.27.1.97 set_tuple_value . . . . .	109
4.27.1.98 set_value_sort . . . . .	109
4.27.1.99 set_variable_hashtable . . . . .	109
4.27.1.100set_variable_hashtables . . . . .	109
4.27.1.101sort_ht_iterator . . . . .	109



4.27.1.102	track_alive	109
4.27.1.103	un_delete_tuple	109
4.27.1.104	un_delete_tuple_old	109
4.27.1.105	update_alive	109
4.27.1.106	update_current_domains	109
4.27.1.107	update_histogram	110
4.27.2	Variable Documentation	110
4.27.2.1	current_markgeneration	110
4.28	hashtable.h File Reference	110
4.28.1	Detailed Description	112
4.28.2	Macro Definition Documentation	113
4.28.2.1	ADD_FOR_PHYSICAL_WIDTH	113
4.28.2.2	DEF_HIGHT	113
4.28.2.3	DELETED	113
4.28.2.4	INF_HIGHT	113
4.28.2.5	REVTIME	113
4.28.2.6	TIMESTP	113
4.28.2.7	TUPMARK	113
4.28.3	Typedef Documentation	113
4.28.3.1	col_card	113
4.28.3.2	common_var	113
4.28.3.3	histogram	113
4.28.3.4	htable	113
4.28.3.5	tuple_ids	113
4.28.4	Function Documentation	113
4.28.4.1	add_name	113
4.28.4.2	add_tuple	113
4.28.4.3	add_tuple_list	113
4.28.4.4	build_histogram	113
4.28.4.5	clear_deleted_rows	113
4.28.4.6	cluster	113
4.28.4.7	cluster_nosort	113
4.28.4.8	cluster_sort	113
4.28.4.9	col_names_to_vars	113
4.28.4.10	compare_clustered	113
4.28.4.11	compare_col_card	113
4.28.4.12	compare_col_name	113
4.28.4.13	compare_scope	113
4.28.4.14	compare_scope_buble	114
4.28.4.15	convert_conflict_to_support2	114

4.28.4.16	convert_conflict_to_support3	114
4.28.4.17	convert_conflict_to_support4	114
4.28.4.18	convert_conflict_to_supportn	114
4.28.4.19	convert_conflict_to_supportn_rec	114
4.28.4.20	copy_col_names	114
4.28.4.21	copy_htable	114
4.28.4.22	create_fill_htable	114
4.28.4.23	delete_tuple	114
4.28.4.24	destroy_histogram	114
4.28.4.25	destroy_htable	114
4.28.4.26	destroy_htable_o	114
4.28.4.27	destroy_int_common_cols_list	114
4.28.4.28	distinct	114
4.28.4.29	do_compare	114
4.28.4.30	do_match	114
4.28.4.31	do_match_next	114
4.28.4.32	estimate_join_project_size	114
4.28.4.33	estimate_join_size	114
4.28.4.34	estimate_join_size_noproject	114
4.28.4.35	estimate_project_size	114
4.28.4.36	filter_domains_with_unary_constraints	114
4.28.4.37	filter_table	114
4.28.4.38	filter_table_with_current_domains	114
4.28.4.39	fprint_ht	114
4.28.4.40	free_tuple_ids	114
4.28.4.41	get_common_cols	114
4.28.4.42	get_domain_product	115
4.28.4.43	get_hashmap_wd	115
4.28.4.44	get_ht_iterator	115
4.28.4.45	get_int_common_col_count	115
4.28.4.46	get_int_common_cols	115
4.28.4.47	get_int_common_cols_list	115
4.28.4.48	get_int_common_cols_vars_list	115
4.28.4.49	get_int_pair	115
4.28.4.50	get_it_object	115
4.28.4.51	get_it_value	115
4.28.4.52	get_next	115
4.28.4.53	get_next_alive_index	115
4.28.4.54	get_next_tuple	115
4.28.4.55	get_projection_scope	115

4.28.4.56	get_tuple	115
4.28.4.57	get_tuple_value	115
4.28.4.58	get_value	115
4.28.4.59	get_value_sort	115
4.28.4.60	has_int_common_cols	115
4.28.4.61	hash_join_tables	115
4.28.4.62	hash_project_distinct_table	115
4.28.4.63	have_value	115
4.28.4.64	init_get_next_tuple	115
4.28.4.65	init_hashmap	115
4.28.4.66	join_tables	115
4.28.4.67	join_tables_new	115
4.28.4.68	join_tables_w_suspend	115
4.28.4.69	load_from_conflict_tuples	116
4.28.4.70	load_from_tuples	116
4.28.4.71	match_tuple_a_tuple_b	116
4.28.4.72	match_tuple_single_ht	116
4.28.4.73	merge	116
4.28.4.74	merge_tuple	116
4.28.4.75	merge_two_tuples_ids	116
4.28.4.76	new_clustered_htable	116
4.28.4.77	new_common_var	116
4.28.4.78	new_histogram	116
4.28.4.79	new_htable	116
4.28.4.80	new_htable_replace	116
4.28.4.81	new_names	116
4.28.4.82	new_tuple_ids	116
4.28.4.83	prepare_pipelined_join	116
4.28.4.84	print_ht	116
4.28.4.85	print_ht_h	116
4.28.4.86	print_ht_line	116
4.28.4.87	print_ht_mysql	116
4.28.4.88	project_distinct_table	116
4.28.4.89	project_join_tables	116
4.28.4.90	project_table	116
4.28.4.91	purge_htable	116
4.28.4.92	rebuild_histogram	116
4.28.4.93	replace_data	116
4.28.4.94	resetset_variable_hashtables	116
4.28.4.95	set_it_object	117

4.28.4.96	set_tuple_value	117
4.28.4.97	set_value_sort	117
4.28.4.98	set_variable_hashtable	117
4.28.4.99	set_variable_hashtables	117
4.28.4.100	sort_ht_iterator	117
4.28.4.101	track_alive	117
4.28.4.102	un_delete_tuple	117
4.28.4.103	un_delete_tuple_old	117
4.28.4.104	update_alive	117
4.28.4.105	update_current_domains	117
4.28.4.106	update_histogram	117
4.29	hashtable_search.c File Reference	117
4.29.1	Function Documentation	118
4.29.1.1	filter_relations_with_1_sol	118
4.29.1.2	filter_this_relation_with_1_sol	118
4.29.1.3	generate_tables_for_scopes_using_relations	118
4.29.1.4	generate_tables_for_scopes_using_relations_noproject	118
4.29.1.5	get_projected_on_variables	118
4.29.1.6	join_filter_relations_with_all_sols	118
4.29.1.7	join_project_multiple_in_multiple_out	118
4.29.1.8	join_project_multiple_in_multiple_out_noextend	118
4.29.1.9	join_project_multiple_in_multiple_out_noextend_given_tables	118
4.29.1.10	process_with_all_sols	118
4.29.1.11	process_with_all_sols_first_var	118
4.29.1.12	project_combine_relations	118
4.29.1.13	project_with_tree	118
4.29.1.14	project_with_tree_payload	118
4.29.1.15	sort_with_tree	118
4.30	hashtable_search.h File Reference	118
4.30.1	Detailed Description	119
4.30.2	Function Documentation	119
4.30.2.1	filter_relations_with_1_sol	119
4.30.2.2	filter_this_relation_with_1_sol	119
4.30.2.3	find_all_supports_with_all_solutions_with_fc_noextend	119
4.30.2.4	generate_tables_for_scopes_using_relations	119
4.30.2.5	generate_tables_for_scopes_using_relations_noproject	119
4.30.2.6	get_projected_on_variables	119
4.30.2.7	join_filter_relations_with_all_sols	119
4.30.2.8	join_project_multiple_in_multiple_out	119
4.30.2.9	join_project_multiple_in_multiple_out_noextend	119

4.30.2.10	<a href="#">join_project_multiple_in_multiple_out_noextend_given_tables</a>	119
4.30.2.11	<a href="#">process_with_all_sols</a>	119
4.30.2.12	<a href="#">process_with_all_sols_first_var</a>	120
4.30.2.13	<a href="#">project_combine_relations</a>	120
4.30.2.14	<a href="#">project_with_tree</a>	120
4.30.2.15	<a href="#">project_with_tree_payload</a>	120
4.30.2.16	<a href="#">sort_with_tree</a>	120
4.31	<a href="#">heap.c File Reference</a>	120
4.31.1	<a href="#">Function Documentation</a>	120
4.31.1.1	<a href="#">build_min_heap</a>	120
4.31.1.2	<a href="#">destroy_heap</a>	120
4.31.1.3	<a href="#">heap_delete</a>	120
4.31.1.4	<a href="#">heap_exchange</a>	120
4.31.1.5	<a href="#">heap_extract_min</a>	120
4.31.1.6	<a href="#">heap_insert</a>	120
4.31.1.7	<a href="#">heap_left</a>	121
4.31.1.8	<a href="#">heap_min</a>	121
4.31.1.9	<a href="#">heap_move_up</a>	121
4.31.1.10	<a href="#">heap_parent</a>	121
4.31.1.11	<a href="#">heap_right</a>	121
4.31.1.12	<a href="#">heap_update</a>	121
4.31.1.13	<a href="#">min_heapify</a>	121
4.31.1.14	<a href="#">new_heap</a>	121
4.31.1.15	<a href="#">print_heap</a>	121
4.32	<a href="#">heap.h File Reference</a>	121
4.32.1	<a href="#">Detailed Description</a>	122
4.32.2	<a href="#">Macro Definition Documentation</a>	122
4.32.2.1	<a href="#">INF</a>	122
4.32.3	<a href="#">Typedef Documentation</a>	122
4.32.3.1	<a href="#">heap</a>	122
4.32.4	<a href="#">Function Documentation</a>	122
4.32.4.1	<a href="#">build_min_heap</a>	122
4.32.4.2	<a href="#">destroy_heap</a>	122
4.32.4.3	<a href="#">heap_delete</a>	122
4.32.4.4	<a href="#">heap_exchange</a>	122
4.32.4.5	<a href="#">heap_extract_min</a>	122
4.32.4.6	<a href="#">heap_insert</a>	122
4.32.4.7	<a href="#">heap_left</a>	122
4.32.4.8	<a href="#">heap_min</a>	122
4.32.4.9	<a href="#">heap_move_up</a>	122

4.32.4.10	heap_parent	122
4.32.4.11	heap_right	122
4.32.4.12	heap_update	122
4.32.4.13	min_heapify	122
4.32.4.14	new_heap	122
4.32.4.15	print_heap	122
4.33	histogram.c File Reference	122
4.34	histogram.h File Reference	123
4.34.1	Detailed Description	123
4.35	llist.c File Reference	123
4.35.1	Function Documentation	124
4.35.1.1	add_int	124
4.35.1.2	add_int_tail	124
4.35.1.3	add_int_tail_distinct	124
4.35.1.4	add_node	124
4.35.1.5	add_node_tail	124
4.35.1.6	add_node_tail_distinct	124
4.35.1.7	add_this_node	124
4.35.1.8	add_this_node_tail	124
4.35.1.9	copy_int_llist_add_pos	124
4.35.1.10	copy_llist	124
4.35.1.11	copy_llist_to	124
4.35.1.12	copy_llist_to_distinct	124
4.35.1.13	copy_value_llist	124
4.35.1.14	destroy_list	124
4.35.1.15	destroy_list_body	124
4.35.1.16	destroy_list_nodes	124
4.35.1.17	destroy_list_nodes_body	124
4.35.1.18	destroy_list_nodes_only	124
4.35.1.19	find_node_of_ivalue	124
4.35.1.20	find_node_of_value	125
4.35.1.21	find_remove_node_of_body	125
4.35.1.22	get_max	125
4.35.1.23	get_min	125
4.35.1.24	get_node_of_body	125
4.35.1.25	get_node_of_ibody	125
4.35.1.26	is_empty	125
4.35.1.27	llist_dequeue	125
4.35.1.28	llist_enqueue	125
4.35.1.29	max_int	125

4.35.1.30	<code>min_int</code>	125
4.35.1.31	<code>move_llist_to</code>	125
4.35.1.32	<code>move_to_head</code>	125
4.35.1.33	<code>move_to_tail</code>	125
4.35.1.34	<code>new_light_stack</code>	125
4.35.1.35	<code>new_light_stack_node</code>	125
4.35.1.36	<code>new_llist</code>	125
4.35.1.37	<code>new_lnode</code>	125
4.35.1.38	<code>peek_s</code>	125
4.35.1.39	<code>pop_light_stack</code>	125
4.35.1.40	<code>pop_s</code>	125
4.35.1.41	<code>push_s</code>	125
4.35.1.42	<code>push_to_light_stack</code>	125
4.35.1.43	<code>remove_head</code>	125
4.35.1.44	<code>remove_max</code>	125
4.35.1.45	<code>remove_min</code>	125
4.35.1.46	<code>remove_node</code>	125
4.35.1.47	<code>remove_node_no_check</code>	125
4.35.1.48	<code>remove_node_no_check_no_null</code>	126
4.35.1.49	<code>remove_node_of_body</code>	126
4.35.1.50	<code>remove_node_of_ibody</code>	126
4.35.1.51	<code>remove_single_node_of_body</code>	126
4.35.1.52	<code>remove_tail</code>	126
4.35.1.53	<code>restore_element_to_same_position</code>	126
4.36	<code>llist.h</code> File Reference	126
4.36.1	Detailed Description	127
4.36.2	Typedef Documentation	127
4.36.2.1	<code>light_stack</code>	127
4.36.2.2	<code>light_stack_node</code>	127
4.36.2.3	<code>llist</code>	127
4.36.2.4	<code>llist_node</code>	127
4.36.3	Function Documentation	127
4.36.3.1	<code>add_int</code>	127
4.36.3.2	<code>add_int_tail</code>	127
4.36.3.3	<code>add_int_tail_distinct</code>	128
4.36.3.4	<code>add_node</code>	128
4.36.3.5	<code>add_node_tail</code>	128
4.36.3.6	<code>add_node_tail_distinct</code>	128
4.36.3.7	<code>add_this_node</code>	128
4.36.3.8	<code>add_this_node_tail</code>	128

4.36.3.9	<a href="#">copy_int_llist_add_pos</a>	128
4.36.3.10	<a href="#">copy_llist</a>	128
4.36.3.11	<a href="#">copy_llist_to</a>	128
4.36.3.12	<a href="#">copy_llist_to_distinct</a>	128
4.36.3.13	<a href="#">copy_value_llist</a>	128
4.36.3.14	<a href="#">destroy_list</a>	128
4.36.3.15	<a href="#">destroy_list_body</a>	128
4.36.3.16	<a href="#">destroy_list_nodes</a>	128
4.36.3.17	<a href="#">destroy_list_nodes_body</a>	128
4.36.3.18	<a href="#">destroy_list_nodes_only</a>	128
4.36.3.19	<a href="#">find_node_of_ivalue</a>	128
4.36.3.20	<a href="#">find_node_of_value</a>	128
4.36.3.21	<a href="#">find_remove_node_of_body</a>	128
4.36.3.22	<a href="#">get_max</a>	128
4.36.3.23	<a href="#">get_min</a>	128
4.36.3.24	<a href="#">get_node_of_body</a>	128
4.36.3.25	<a href="#">get_node_of_ibody</a>	128
4.36.3.26	<a href="#">is_empty</a>	128
4.36.3.27	<a href="#">llist_dequeue</a>	128
4.36.3.28	<a href="#">llist_enqueue</a>	128
4.36.3.29	<a href="#">max_int</a>	128
4.36.3.30	<a href="#">min_int</a>	128
4.36.3.31	<a href="#">move_llist_to</a>	129
4.36.3.32	<a href="#">move_to_head</a>	129
4.36.3.33	<a href="#">move_to_tail</a>	129
4.36.3.34	<a href="#">new_light_stack</a>	129
4.36.3.35	<a href="#">new_light_stack_node</a>	129
4.36.3.36	<a href="#">new_llist</a>	129
4.36.3.37	<a href="#">new_lnode</a>	129
4.36.3.38	<a href="#">peek_s</a>	129
4.36.3.39	<a href="#">pop_light_stack</a>	129
4.36.3.40	<a href="#">pop_s</a>	129
4.36.3.41	<a href="#">push_s</a>	129
4.36.3.42	<a href="#">push_to_light_stack</a>	129
4.36.3.43	<a href="#">remove_head</a>	129
4.36.3.44	<a href="#">remove_max</a>	129
4.36.3.45	<a href="#">remove_min</a>	129
4.36.3.46	<a href="#">remove_node</a>	129
4.36.3.47	<a href="#">remove_node_no_check</a>	129
4.36.3.48	<a href="#">remove_node_no_check_no_null</a>	129



4.36.3.49	remove_node_of_body	129
4.36.3.50	remove_node_of_ibody	129
4.36.3.51	remove_single_node_of_body	129
4.36.3.52	remove_tail	129
4.36.3.53	restore_element_to_same_position	129
4.37	main.c File Reference	130
4.37.1	Detailed Description	130
4.37.2	Function Documentation	133
4.37.2.1	check_ok	133
4.37.2.2	check_ok2	133
4.37.2.3	main	133
4.37.2.4	sigxcpu_handler	133
4.37.3	Variable Documentation	133
4.37.3.1	m_s_g	133
4.38	minibucket.c File Reference	133
4.38.1	Function Documentation	133
4.38.1.1	mbe_join_tables	134
4.38.1.2	partition_bucket	134
4.39	minibucket.h File Reference	134
4.39.1	Detailed Description	134
4.39.2	Function Documentation	134
4.39.2.1	mbe_join_tables	134
4.39.2.2	partition_bucket	134
4.39.2.3	partition_bucket_old	134
4.40	oneall.c File Reference	134
4.40.1	Function Documentation	134
4.40.1.1	evaluate_on_tree	134
4.40.1.2	get_dt_node	134
4.40.1.3	parse_tree	134
4.40.1.4	print_decision_tree	134
4.41	oneall.h File Reference	135
4.41.1	Detailed Description	135
4.41.2	Typedef Documentation	135
4.41.2.1	decision_tree	135
4.41.2.2	decision_tree_system	135
4.41.3	Function Documentation	135
4.41.3.1	evaluate_on_tree	135
4.41.3.2	get_dt_node	135
4.41.3.3	parse_tree	135
4.41.3.4	print_decision_tree	135

4.42	parsers.c File Reference	135
4.42.1	Function Documentation	136
4.42.1.1	read_comment	136
4.42.1.2	read_ergos	136
4.42.1.3	read_uai	136
4.42.1.4	read_wcsp	136
4.42.1.5	unrank	136
4.43	parsers.h File Reference	136
4.43.1	Detailed Description	136
4.43.2	Function Documentation	136
4.43.2.1	read_comment	136
4.43.2.2	read_ergos	136
4.43.2.3	read_uai	136
4.43.2.4	read_wcsp	136
4.43.2.5	unrank	136
4.44	partition.c File Reference	137
4.44.1	Function Documentation	137
4.44.1.1	all_binary_rel_separator	137
4.44.1.2	binary_rel_separator	137
4.44.1.3	partition_separator	137
4.45	partition.h File Reference	137
4.45.1	Detailed Description	137
4.45.2	Function Documentation	137
4.45.2.1	all_binary_rel_separator	137
4.45.2.2	binary_rel_separator	137
4.45.2.3	partition_separator	137
4.46	query_graph.c File Reference	138
4.46.1	Function Documentation	139
4.46.1.1	add_constraint_to_graph	139
4.46.1.2	add_constraint_to_graph_force	139
4.46.1.3	add_constraints_to_graph	139
4.46.1.4	add_neighbours_for_fill_edge	139
4.46.1.5	add_neighbours_for_fill_edge_given_constraints_set	139
4.46.1.6	add_neighbours_for_node	139
4.46.1.7	add_neighbours_for_node_given_constraint_set	140
4.46.1.8	add_or_get_constraint_node	140
4.46.1.9	append_to_nodes_combination	140
4.46.1.10	build_adjacency_matrix	140
4.46.1.11	build_adjacency_matrix2	140
4.46.1.12	build_plan	140

4.46.1.13	check_connected	140
4.46.1.14	compare	140
4.46.1.15	compare_scope_cge	140
4.46.1.16	compare_scope_cge_global_list	140
4.46.1.17	delete_node	140
4.46.1.18	destroy_constraint_graph	140
4.46.1.19	destroy_constraint_graph_edge	140
4.46.1.20	destroy_constraint_graph_node	140
4.46.1.21	evaluate_edges	140
4.46.1.22	expand_cgn_i	140
4.46.1.23	get_all_nodes_combination	140
4.46.1.24	get_cnode_id	140
4.46.1.25	get_constraint_graph_node	140
4.46.1.26	get_edge	140
4.46.1.27	get_estimated_value_weight	140
4.46.1.28	get_initial_node	140
4.46.1.29	get_node_with_constraint_id	140
4.46.1.30	get_object_edge_sort	140
4.46.1.31	get_other_end	140
4.46.1.32	get_value_edge_sort	140
4.46.1.33	has_common_cols	140
4.46.1.34	init_graph	141
4.46.1.35	init_graph_given_constraint_set	141
4.46.1.36	init_graph_with_fill_edges	141
4.46.1.37	measure	141
4.46.1.38	merge_nodes	141
4.46.1.39	merge_nodes_no_join	141
4.46.1.40	new_constraint_graph	141
4.46.1.41	new_constraint_graph_edge	141
4.46.1.42	new_constraint_graph_node	141
4.46.1.43	new_constraint_graph_node_t	141
4.46.1.44	new_neighbour_set	141
4.46.1.45	print_graph	141
4.46.1.46	print_graph_scopes	141
4.46.1.47	print_number_of_deleted_tuples	141
4.46.1.48	print_tree	141
4.46.1.49	printGraphNeato	141
4.46.1.50	printGraphNeato2	141
4.46.1.51	remove_edge_from_graph	141
4.46.1.52	remove_fill_edges_from_variables	141

4.46.1.53	<a href="#">remove_node_from_graph</a>	141
4.46.1.54	<a href="#">reset_colors</a>	141
4.46.1.55	<a href="#">set_object_edge_sort</a>	141
4.46.1.56	<a href="#">set_weights</a>	141
4.46.1.57	<a href="#">update_constraint_tuples</a>	141
4.46.2	<a href="#">Variable Documentation</a>	141
4.46.2.1	<a href="#">m_s_g</a>	141
4.47	<a href="#">query_graph.h File Reference</a>	142
4.47.1	<a href="#">Detailed Description</a>	143
4.47.2	<a href="#">Macro Definition Documentation</a>	144
4.47.2.1	<a href="#">INITIAL_LIMIT</a>	144
4.47.3	<a href="#">Typedef Documentation</a>	144
4.47.3.1	<a href="#">constraint_graph</a>	144
4.47.3.2	<a href="#">constraint_graph_edge</a>	144
4.47.3.3	<a href="#">constraint_graph_node</a>	144
4.47.4	<a href="#">Function Documentation</a>	144
4.47.4.1	<a href="#">add_constraint_to_graph</a>	144
4.47.4.2	<a href="#">add_constraint_to_graph_force</a>	144
4.47.4.3	<a href="#">add_constraints_to_graph</a>	144
4.47.4.4	<a href="#">add_neighbours_for_fill_edge</a>	144
4.47.4.5	<a href="#">add_neighbours_for_fill_edge_given_constraints_set</a>	144
4.47.4.6	<a href="#">add_neighbours_for_node</a>	144
4.47.4.7	<a href="#">add_neighbours_for_node_given_constraint_set</a>	144
4.47.4.8	<a href="#">add_or_get_constraint_node</a>	144
4.47.4.9	<a href="#">append_to_nodes_combination</a>	144
4.47.4.10	<a href="#">build_adjacency_matrix</a>	144
4.47.4.11	<a href="#">build_adjacency_matrix2</a>	144
4.47.4.12	<a href="#">build_plan</a>	144
4.47.4.13	<a href="#">buld_join_tree</a>	144
4.47.4.14	<a href="#">check_connected</a>	144
4.47.4.15	<a href="#">compare</a>	144
4.47.4.16	<a href="#">compare_scope_cge</a>	144
4.47.4.17	<a href="#">compare_scope_cge_global_list</a>	144
4.47.4.18	<a href="#">delete_node</a>	144
4.47.4.19	<a href="#">destroy_constraint_graph</a>	144
4.47.4.20	<a href="#">destroy_constraint_graph_edge</a>	144
4.47.4.21	<a href="#">destroy_constraint_graph_node</a>	145
4.47.4.22	<a href="#">do_db</a>	145
4.47.4.23	<a href="#">evaluate_edges</a>	145
4.47.4.24	<a href="#">expand_cgn_i</a>	145

4.47.4.25	<a href="#">get_all_nodes_combination</a>	145
4.47.4.26	<a href="#">get_cnode_id</a>	145
4.47.4.27	<a href="#">get_constraint_graph_node</a>	145
4.47.4.28	<a href="#">get_edge</a>	145
4.47.4.29	<a href="#">get_estimated_value_weight</a>	145
4.47.4.30	<a href="#">get_initial_node</a>	145
4.47.4.31	<a href="#">get_int_common_cols</a>	145
4.47.4.32	<a href="#">get_node_with_constraint_id</a>	145
4.47.4.33	<a href="#">get_object_edge_sort</a>	145
4.47.4.34	<a href="#">get_other_end</a>	145
4.47.4.35	<a href="#">get_value_edge_sort</a>	145
4.47.4.36	<a href="#">greedy_naive</a>	145
4.47.4.37	<a href="#">has_common_cols</a>	145
4.47.4.38	<a href="#">init_graph</a>	145
4.47.4.39	<a href="#">init_graph_given_constraint_set</a>	145
4.47.4.40	<a href="#">init_graph_with_fill_edges</a>	145
4.47.4.41	<a href="#">make_tree</a>	145
4.47.4.42	<a href="#">measure</a>	145
4.47.4.43	<a href="#">merge_nodes</a>	145
4.47.4.44	<a href="#">merge_nodes_no_join</a>	145
4.47.4.45	<a href="#">new_constraint_graph</a>	145
4.47.4.46	<a href="#">new_constraint_graph_edge</a>	146
4.47.4.47	<a href="#">new_constraint_graph_node</a>	146
4.47.4.48	<a href="#">new_constraint_graph_node_t</a>	146
4.47.4.49	<a href="#">new_neighbour_set</a>	146
4.47.4.50	<a href="#">print_graph</a>	146
4.47.4.51	<a href="#">print_graph_scopes</a>	146
4.47.4.52	<a href="#">print_number_of_deleted_tuples</a>	146
4.47.4.53	<a href="#">print_tree</a>	146
4.47.4.54	<a href="#">printGraphNeato</a>	146
4.47.4.55	<a href="#">printGraphNeato2</a>	146
4.47.4.56	<a href="#">remove_edge_from_graph</a>	146
4.47.4.57	<a href="#">remove_fill_edges_from_variables</a>	146
4.47.4.58	<a href="#">remove_node_from_graph</a>	146
4.47.4.59	<a href="#">reset_colors</a>	146
4.47.4.60	<a href="#">set_object_edge_sort</a>	146
4.47.4.61	<a href="#">set_weights</a>	146
4.47.4.62	<a href="#">update_constraint_tuples</a>	146
4.48	<a href="#">rb_tree.c File Reference</a>	146
4.48.1	<a href="#">Function Documentation</a>	147

4.48.1.1	<a href="#">check_tree</a>	147
4.48.1.2	<a href="#">destroy_rb_tree</a>	147
4.48.1.3	<a href="#">destroy_rb_tree_key</a>	147
4.48.1.4	<a href="#">destroy_rb_tree_paload</a>	147
4.48.1.5	<a href="#">new_rb_tree</a>	147
4.48.1.6	<a href="#">rb_copy_tree</a>	147
4.48.1.7	<a href="#">rb_delete</a>	147
4.48.1.8	<a href="#">rb_delete_fixup</a>	147
4.48.1.9	<a href="#">rb_delete_key</a>	147
4.48.1.10	<a href="#">rb_find</a>	147
4.48.1.11	<a href="#">rb_get_sorted_order</a>	147
4.48.1.12	<a href="#">rb_get_sorted_order_list</a>	147
4.48.1.13	<a href="#">rb_insert</a>	147
4.48.1.14	<a href="#">rb_insert_fixup</a>	147
4.48.1.15	<a href="#">rb_left_rotate</a>	147
4.48.1.16	<a href="#">rb_maximum</a>	147
4.48.1.17	<a href="#">rb_minimum</a>	147
4.48.1.18	<a href="#">rb_print_tree</a>	147
4.48.1.19	<a href="#">rb_print_tree_l</a>	147
4.48.1.20	<a href="#">rb_right_rotate</a>	147
4.48.1.21	<a href="#">rb_successor</a>	147
4.48.1.22	<a href="#">rb_tree_union</a>	147
4.49	<a href="#">rb_tree.h File Reference</a>	148
4.49.1	<a href="#">Detailed Description</a>	148
4.49.2	<a href="#">Typedef Documentation</a>	148
4.49.2.1	<a href="#">rb_tree</a>	148
4.49.3	<a href="#">Function Documentation</a>	148
4.49.3.1	<a href="#">check_tree</a>	148
4.49.3.2	<a href="#">destroy_rb_tree</a>	149
4.49.3.3	<a href="#">destroy_rb_tree_key</a>	149
4.49.3.4	<a href="#">destroy_rb_tree_paload</a>	149
4.49.3.5	<a href="#">new_rb_tree</a>	149
4.49.3.6	<a href="#">rb_copy_tree</a>	149
4.49.3.7	<a href="#">rb_delete</a>	149
4.49.3.8	<a href="#">rb_delete_fixup</a>	149
4.49.3.9	<a href="#">rb_delete_key</a>	149
4.49.3.10	<a href="#">rb_find</a>	149
4.49.3.11	<a href="#">rb_get_sorted_order</a>	149
4.49.3.12	<a href="#">rb_get_sorted_order_list</a>	149
4.49.3.13	<a href="#">rb_insert</a>	149

4.49.3.14	<code>rb_insert_fixup</code>	149
4.49.3.15	<code>rb_left_rotate</code>	149
4.49.3.16	<code>rb_maximum</code>	149
4.49.3.17	<code>rb_minimum</code>	149
4.49.3.18	<code>rb_print_tree</code>	149
4.49.3.19	<code>rb_print_tree_l</code>	149
4.49.3.20	<code>rb_right_rotate</code>	149
4.49.3.21	<code>rb_successor</code>	149
4.49.3.22	<code>rb_tree_union</code>	149
4.50	<code>rbt_set.c</code> File Reference	149
4.50.1	Function Documentation	150
4.50.1.1	<code>add_element_rbt</code>	150
4.50.1.2	<code>add_int_element_rbt</code>	150
4.50.1.3	<code>copy_set_rbt</code>	150
4.50.1.4	<code>destroy_set_rbt</code>	150
4.50.1.5	<code>get_element_with_key_rbt</code>	150
4.50.1.6	<code>get_int_element_rbt</code>	150
4.50.1.7	<code>new_rbt_set</code>	150
4.50.1.8	<code>remove_element_with_key_rbt</code>	150
4.50.1.9	<code>remove_int_element_rbt</code>	150
4.50.1.10	<code>reset_set_rbt</code>	150
4.50.1.11	<code>set_empty_rbt</code>	150
4.50.1.12	<code>set_to_list_rbt</code>	150
4.51	<code>rbt_set.h</code> File Reference	150
4.51.1	Detailed Description	151
4.51.2	Typedef Documentation	151
4.51.2.1	<code>rbt_set</code>	151
4.51.3	Function Documentation	151
4.51.3.1	<code>add_element_rbt</code>	151
4.51.3.2	<code>add_int_element_rbt</code>	151
4.51.3.3	<code>copy_set_rbt</code>	151
4.51.3.4	<code>destroy_set_rbt</code>	151
4.51.3.5	<code>get_element_with_key_rbt</code>	151
4.51.3.6	<code>get_int_element_rbt</code>	151
4.51.3.7	<code>new_rbt_set</code>	151
4.51.3.8	<code>peek_element_rbt</code>	151
4.51.3.9	<code>remove_element_with_key_rbt</code>	151
4.51.3.10	<code>remove_int_element_rbt</code>	151
4.51.3.11	<code>reset_set_rbt</code>	151
4.51.3.12	<code>set_empty_rbt</code>	151

4.51.3.13 set_to_list_rbt . . . . .	151
4.52 rpwc.c File Reference . . . . .	151
4.52.1 Function Documentation . . . . .	152
4.52.1.1 destroy_lastGAC . . . . .	152
4.52.1.2 destroy_undo_info . . . . .	152
4.52.1.3 domain_filter . . . . .	152
4.52.1.4 enqueue . . . . .	152
4.52.1.5 init_rpwc . . . . .	152
4.52.1.6 new_lastGAC . . . . .	152
4.52.1.7 new_tripplet . . . . .	152
4.52.1.8 new_undo_info . . . . .	152
4.52.1.9 new_undo_lastGAC . . . . .	152
4.52.1.10 reviseGAC . . . . .	152
4.52.1.11 reviseGACMaxRPWC . . . . .	152
4.52.1.12 reviseMaxRPWC . . . . .	152
4.53 rpwc.h File Reference . . . . .	152
4.53.1 Detailed Description . . . . .	153
4.53.2 Typedef Documentation . . . . .	153
4.53.2.1 lastGAC . . . . .	153
4.53.2.2 triplet . . . . .	153
4.53.2.3 undo_info . . . . .	153
4.53.2.4 undo_lastGAC . . . . .	153
4.53.3 Enumeration Type Documentation . . . . .	153
4.53.3.1 reviseAlgo . . . . .	153
4.53.4 Function Documentation . . . . .	154
4.53.4.1 destroy_lastGAC . . . . .	154
4.53.4.2 destroy_undo_info . . . . .	154
4.53.4.3 domain_filter . . . . .	154
4.53.4.4 enqueue . . . . .	154
4.53.4.5 init_rpwc . . . . .	154
4.53.4.6 new_lastGAC . . . . .	154
4.53.4.7 new_tripplet . . . . .	154
4.53.4.8 new_undo_info . . . . .	154
4.53.4.9 new_undo_lastGAC . . . . .	154
4.53.4.10 reviseGAC . . . . .	154
4.53.4.11 reviseGACMaxRPWC . . . . .	154
4.53.4.12 reviseMaxRPWC . . . . .	154
4.54 rr.c File Reference . . . . .	154
4.54.1 Function Documentation . . . . .	154
4.54.1.1 add_rr_edges . . . . .	154



---

4.54.1.2	get_overlaps	155
4.54.1.3	get_relation_groups	155
4.54.1.4	rr	155
4.55	rr.h File Reference	155
4.55.1	Detailed Description	155
4.55.2	Function Documentation	155
4.55.2.1	add_rr_edges	155
4.55.2.2	get_overlaps	155
4.55.2.3	get_relation_groups	155
4.55.2.4	rr	155
4.56	rs2.c File Reference	155
4.56.1	Function Documentation	157
4.56.1.1	build_ind_trees	157
4.56.1.2	choose_var	157
4.56.1.3	choose_var_priority	157
4.56.1.4	choose_var_priority_decided	157
4.56.1.5	clear_comb_datas_from_hts	157
4.56.1.6	destroy_comb	157
4.56.1.7	destroy_comb_cdata	157
4.56.1.8	destroy_comb_data2	157
4.56.1.9	enum_table_pairs	157
4.56.1.10	filter_all_relations_with_all_solutions	157
4.56.1.11	filter_all_relations_with_all_solutions_priority	158
4.56.1.12	filter_single_relation_in_combination	158
4.56.1.13	find_all_supports_with_all_solutions_with_fc	158
4.56.1.14	find_all_supports_with_all_solutions_with_fc_priority	158
4.56.1.15	find_all_supports_with_all_solutions_with_fc_priority_next_var	158
4.56.1.16	find_support_for_single_solution	158
4.56.1.17	find_support_for_single_solution_bc	158
4.56.1.18	find_support_for_single_solution_bc_rec	158
4.56.1.19	find_support_for_single_solution_with_fc	158
4.56.1.20	find_support_for_single_solution_with_fc_priority	158
4.56.1.21	find_support_for_single_solution_with_fc_rec	158
4.56.1.22	find_support_for_single_solution_with_fc_rec_single_out	158
4.56.1.23	get_comb_from	158
4.56.1.24	get_ht_from_cd	158
4.56.1.25	get_hts_paired	158
4.56.1.26	get_pair_common_cols	158
4.56.1.27	get_revision_time	158
4.56.1.28	get_rstat	158

---

4.56.1.29	<code>get_tms_paired</code>	158
4.56.1.30	<code>make_comb_data2</code>	158
4.56.1.31	<code>mark_all_in_future</code>	159
4.56.1.32	<code>new_comb_cdata</code>	159
4.56.1.33	<code>queue_relation_related</code>	159
4.56.1.34	<code>queue_relation_related_multiqueue</code>	159
4.56.1.35	<code>set_deg_count</code>	159
4.56.1.36	<code>set_dom</code>	159
4.56.1.37	<code>set_revision_time</code>	159
4.56.1.38	<code>set_revision_time_parents</code>	159
4.56.1.39	<code>setup_relation_group_for_a_comb</code>	159
4.56.1.40	<code>setup_relation_group_for_a_comb_first</code>	159
4.56.1.41	<code>setup_relation_group_for_a_comb_long_indexes</code>	159
4.56.1.42	<code>setup_relation_groups</code>	159
4.56.1.43	<code>setup_relation_groups_</code>	159
4.56.1.44	<code>stage4_filter</code>	159
4.56.1.45	<code>stage4_filter_auto</code>	159
4.56.1.46	<code>stage4_filter_auto_all_sol</code>	159
4.56.1.47	<code>stage4_filter_auto_one_sol</code>	159
4.56.1.48	<code>stage4_filter_multiqueue</code>	159
4.56.1.49	<code>stage4_filter_with_all_solutions</code>	159
4.56.1.50	<code>stage4_filter_with_no_check_insol_tuples</code>	159
4.56.1.51	<code>stage4_init_queue</code>	159
4.56.1.52	<code>undo_dom_deg_var</code>	159
4.56.1.53	<code>visit_rest_of_tuples</code>	159
4.56.2	Variable Documentation	160
4.56.2.1	<code>watch</code>	160
4.57	<code>rs2.h</code> File Reference	160
4.57.1	Detailed Description	162
4.57.2	Typedef Documentation	162
4.57.2.1	<code>comb_cdata</code>	162
4.57.2.2	<code>table_map</code>	162
4.57.3	Function Documentation	162
4.57.3.1	<code>build_ind_trees</code>	162
4.57.3.2	<code>build_joining_order</code>	162
4.57.3.3	<code>choose_var</code>	162
4.57.3.4	<code>choose_var_priority</code>	162
4.57.3.5	<code>choose_var_priority_decided</code>	162
4.57.3.6	<code>clear_comb_datas_from_hts</code>	162
4.57.3.7	<code>clear_support_tuple_sets</code>	162

4.57.3.8	destroy_comb	162
4.57.3.9	destroy_comb_cdata	162
4.57.3.10	destroy_comb_data2	162
4.57.3.11	destroy_support	162
4.57.3.12	enum_table_pairs	162
4.57.3.13	filter_all_relations_with_all_solutions	162
4.57.3.14	filter_all_relations_with_all_solutions_priority	163
4.57.3.15	filter_relations_rs	163
4.57.3.16	filter_single_relation_in_combination	163
4.57.3.17	find_all_supports_with_all_solutions_with_fc	163
4.57.3.18	find_all_supports_with_all_solutions_with_fc_priority	163
4.57.3.19	find_all_supports_with_all_solutions_with_fc_priority_next_var	163
4.57.3.20	find_support_for_single_solution	163
4.57.3.21	find_support_for_single_solution_bc	163
4.57.3.22	find_support_for_single_solution_bc_rec	163
4.57.3.23	find_support_for_single_solution_rec	163
4.57.3.24	find_support_for_single_solution_with_fc	163
4.57.3.25	find_support_for_single_solution_with_fc_priority	163
4.57.3.26	find_support_for_single_solution_with_fc_rec	163
4.57.3.27	find_support_for_single_solution_with_fc_rec_single_out	163
4.57.3.28	get_comb_from	163
4.57.3.29	get_ht_from_cd	163
4.57.3.30	get_hts_paired	163
4.57.3.31	get_pair_common_cols	163
4.57.3.32	get_revision_time	163
4.57.3.33	get_rstat	163
4.57.3.34	get_tms_paired	163
4.57.3.35	make_comb_data2	164
4.57.3.36	mark_all_in_future	164
4.57.3.37	new_comb_cdata	164
4.57.3.38	new_support	164
4.57.3.39	new_update_support	164
4.57.3.40	new_updated_combinations	164
4.57.3.41	queue_related_relations	164
4.57.3.42	queue_relation_related	164
4.57.3.43	queue_relation_related_multiqueue	164
4.57.3.44	queue_supported_tuples	164
4.57.3.45	remove_changed_supports	164
4.57.3.46	set_deg_count	164
4.57.3.47	set_dom	164

4.57.3.48	set_revision_time	164
4.57.3.49	set_revision_time_parents	164
4.57.3.50	setup_relation_group_for_a_comb	164
4.57.3.51	setup_relation_group_for_a_comb_first	164
4.57.3.52	setup_relation_group_for_a_comb_long_indexes	164
4.57.3.53	setup_relation_groups	164
4.57.3.54	setup_relation_groups_	164
4.57.3.55	stage4_filter	164
4.57.3.56	stage4_filter_auto	164
4.57.3.57	stage4_filter_auto_all_sol	164
4.57.3.58	stage4_filter_auto_one_sol	164
4.57.3.59	stage4_filter_multiqueue	165
4.57.3.60	stage4_filter_with_all_solutions	165
4.57.3.61	stage4_filter_with_no_check_insol_tuples	165
4.57.3.62	stage4_init_queue	165
4.57.3.63	undo_dom_deg_var	165
4.57.3.64	visit_rest_of_tuples	165
4.58	rs2tree.c File Reference	165
4.58.1	Function Documentation	167
4.58.1.1	add_times_to_stats	167
4.58.1.2	assign_table_context	167
4.58.1.3	assign_variable	167
4.58.1.4	avgStdev	167
4.58.1.5	avgStdev2	167
4.58.1.6	check_this	167
4.58.1.7	choose_var_from_treen	167
4.58.1.8	extract_feature	167
4.58.1.9	filter_table_with_chosen_value	167
4.58.1.10	find_top_root_with_scope	167
4.58.1.11	fringe_extract_top	167
4.58.1.12	get_make_stats	167
4.58.1.13	get_payload	167
4.58.1.14	init_processed_info	167
4.58.1.15	init_update_processed_info	167
4.58.1.16	new_rs2tree_stats	167
4.58.1.17	post_feature_extract	167
4.58.1.18	pre_feature_extract	168
4.58.1.19	print_rs2tree_stats	168
4.58.1.20	process_geneate_bolstering_bin_clique_relations	168
4.58.1.21	process_this_cluster	168

4.58.1.22 process_tree_combinations	168
4.58.1.23 process_tree_combinations2	168
4.58.1.24 process_tree_combinations3	168
4.58.1.25 process_tree_combinations4	168
4.58.1.26 process_tree_combinations4_	168
4.58.1.27 process_tree_combinations4_filter	168
4.58.1.28 process_tree_combinations4_filter_down_rec	168
4.58.1.29 process_tree_combinations4_filter_up_rec	168
4.58.1.30 process_tree_combinations5	168
4.58.1.31 process_tree_combinations52	168
4.58.1.32 process_tree_combinations522	168
4.58.1.33 process_tree_combinations53	168
4.58.1.34 process_tree_combinations54	168
4.58.1.35 process_tree_combinations6	168
4.58.1.36 process_tree_combinations7	168
4.58.1.37 process_tree_combinations8	169
4.58.1.38 process_tree_combinations9	169
4.58.1.39 record_good_nogood	169
4.58.1.40 report_tree	169
4.58.1.41 reset_relation_queues	169
4.58.1.42 reset_variable_domain	169
4.58.1.43 reset_variable_domain2	169
4.58.1.44 rs2_all_sols_bt	169
4.58.1.45 rs2_all_sols_bt_pre_processing	169
4.58.1.46 rs2t_get_heap_value	169
4.58.1.47 rs2t_get_id	169
4.58.1.48 rs2t_set_heap_pos	169
4.58.1.49 rs2t_set_heap_value	169
4.58.1.50 rs2tree_filter_domains	169
4.58.1.51 rs2tree_filter_domains_given_relation	169
4.58.1.52 rs2tree_filter_domains_given_relations	169
4.58.1.53 rs2tree_filter_relations_given_domains	169
4.58.1.54 set_cluster_constraints_to_projected_reols_on_c_vars	169
4.58.1.55 set_default_context_to_tables	169
4.58.1.56 set_good_nogood_trees	169
4.58.1.57 set_queues_treenodes2	169
4.58.1.58 set_queues_treenodes2_singleq	169
4.58.1.59 set_single_graph_combinations_to_treenodes	169
4.58.1.60 set_variable_htables2	169
4.58.1.61 single_graph_combinations_with_tables	170

4.58.1.62	start_w_b_BT D	170
4.58.1.63	startBT D	170
4.58.1.64	traverse_treemap_set_collect_counts	170
4.58.1.65	un_assign_variable	170
4.58.1.66	undo_tuple_deletions_for_rs2tree	170
4.58.1.67	undo_value_deletions	170
4.58.1.68	update_processed_info	170
4.59	rs2tree.h File Reference	170
4.59.1	Detailed Description	172
4.59.2	Typedef Documentation	172
4.59.2.1	rs2tree_configuration	172
4.59.2.2	rs2tree_stats	172
4.59.3	Function Documentation	172
4.59.3.1	add_times_to_stats	172
4.59.3.2	assign_table_context	172
4.59.3.3	assign_variable	172
4.59.3.4	avgStdev	172
4.59.3.5	avgStdev2	172
4.59.3.6	check_this	172
4.59.3.7	choose_var_from_treen	172
4.59.3.8	collect_htables	173
4.59.3.9	extract_feature	173
4.59.3.10	filter_table_with_chosen_value	173
4.59.3.11	find_top_root_with_scope	173
4.59.3.12	fringe_extract_top	173
4.59.3.13	generate_combinations_per_cluster	173
4.59.3.14	get_make_stats	173
4.59.3.15	get_payload	173
4.59.3.16	init_processed_info	173
4.59.3.17	init_update_processed_info	173
4.59.3.18	new_rs2tree_stats	173
4.59.3.19	post_feature_extract	173
4.59.3.20	pre_feature_extract	173
4.59.3.21	print_rs2tree_stats	173
4.59.3.22	process_geneate_bolstering_bin_clique_relations	173
4.59.3.23	process_this_cluster	173
4.59.3.24	process_tree_combinations	173
4.59.3.25	process_tree_combinations2	173
4.59.3.26	process_tree_combinations3	173
4.59.3.27	process_tree_combinations4	173

4.59.3.28 process_tree_combinations4_ . . . . .	173
4.59.3.29 process_tree_combinations4_filter . . . . .	173
4.59.3.30 process_tree_combinations4_filter_down_rec . . . . .	173
4.59.3.31 process_tree_combinations4_filter_up_rec . . . . .	174
4.59.3.32 process_tree_combinations5 . . . . .	174
4.59.3.33 process_tree_combinations52 . . . . .	174
4.59.3.34 process_tree_combinations522 . . . . .	174
4.59.3.35 process_tree_combinations53 . . . . .	174
4.59.3.36 process_tree_combinations54 . . . . .	174
4.59.3.37 process_tree_combinations6 . . . . .	174
4.59.3.38 process_tree_combinations7 . . . . .	174
4.59.3.39 process_tree_combinations8 . . . . .	174
4.59.3.40 process_tree_combinations9 . . . . .	174
4.59.3.41 record_good_nogood . . . . .	174
4.59.3.42 report_tree . . . . .	174
4.59.3.43 reset_relation_queues . . . . .	174
4.59.3.44 reset_variable_domain . . . . .	174
4.59.3.45 reset_variable_domain2 . . . . .	174
4.59.3.46 rs2_all_sols_bt . . . . .	174
4.59.3.47 rs2_all_sols_bt_pre_processing . . . . .	174
4.59.3.48 rs2t_get_heap_value . . . . .	174
4.59.3.49 rs2t_get_id . . . . .	174
4.59.3.50 rs2t_set_heap_pos . . . . .	174
4.59.3.51 rs2t_set_heap_value . . . . .	175
4.59.3.52 rs2tree_filter_domains . . . . .	175
4.59.3.53 rs2tree_filter_domains_given_relation . . . . .	175
4.59.3.54 rs2tree_filter_domains_given_relations . . . . .	175
4.59.3.55 rs2tree_filter_relations_given_domains . . . . .	175
4.59.3.56 set_cluster_constraints_to_projected_reols_on_c_vars . . . . .	175
4.59.3.57 set_combinations . . . . .	175
4.59.3.58 set_combinations2 . . . . .	175
4.59.3.59 set_combinations3 . . . . .	175
4.59.3.60 set_default_context_to_tables . . . . .	175
4.59.3.61 set_good_nogood_trees . . . . .	175
4.59.3.62 set_queues_treenodes . . . . .	175
4.59.3.63 set_queues_treenodes2 . . . . .	175
4.59.3.64 set_queues_treenodes2_singleq . . . . .	175
4.59.3.65 set_single_graph_combinations_to_treenodes . . . . .	175
4.59.3.66 set_variable_htables . . . . .	175
4.59.3.67 set_variable_htables2 . . . . .	175

4.59.3.68	<a href="#">single_graph_combinations_with_tables</a>	175
4.59.3.69	<a href="#">start_w_b_BTD</a>	175
4.59.3.70	<a href="#">startBTD</a>	175
4.59.3.71	<a href="#">traverse_treemap_set_collect_counts</a>	175
4.59.3.72	<a href="#">un_assign_variable</a>	175
4.59.3.73	<a href="#">undo_tuple_deletions_for_rs2tree</a>	175
4.59.3.74	<a href="#">undo_value_deletions</a>	175
4.59.3.75	<a href="#">update_processed_info</a>	175
4.60	<a href="#">septables.c File Reference</a>	176
4.60.1	<a href="#">Function Documentation</a>	176
4.60.1.1	<a href="#">add_binary_scopes</a>	176
4.60.1.2	<a href="#">add_maxclique_scopes</a>	176
4.60.1.3	<a href="#">normalize_generate_separator_scopes_relations</a>	176
4.60.1.4	<a href="#">normalize_small_big_lists</a>	176
4.60.1.5	<a href="#">shape_constraint_scopes_pE</a>	176
4.60.1.6	<a href="#">shape_constraint_scopes_pE_global</a>	176
4.60.2	<a href="#">Variable Documentation</a>	176
4.60.2.1	<a href="#">varmap_s</a>	176
4.61	<a href="#">septables.h File Reference</a>	176
4.61.1	<a href="#">Detailed Description</a>	177
4.61.2	<a href="#">Function Documentation</a>	177
4.61.2.1	<a href="#">add_binary_scopes</a>	177
4.61.2.2	<a href="#">add_maxclique_scopes</a>	177
4.61.2.3	<a href="#">normalize_generate_separator_scopes_relations</a>	177
4.61.2.4	<a href="#">normalize_small_big_lists</a>	177
4.61.2.5	<a href="#">shape_constraint_scopes_pE</a>	177
4.61.2.6	<a href="#">shape_constraint_scopes_pE_global</a>	177
4.62	<a href="#">set.c File Reference</a>	177
4.62.1	<a href="#">Function Documentation</a>	178
4.62.1.1	<a href="#">add_element</a>	178
4.62.1.2	<a href="#">add_element_node</a>	178
4.62.1.3	<a href="#">add_element_nomap</a>	178
4.62.1.4	<a href="#">add_int_element</a>	178
4.62.1.5	<a href="#">copy_set</a>	178
4.62.1.6	<a href="#">destroy_set</a>	178
4.62.1.7	<a href="#">get_element_with_key</a>	178
4.62.1.8	<a href="#">get_int_element</a>	178
4.62.1.9	<a href="#">new_s_node</a>	178
4.62.1.10	<a href="#">new_set</a>	178
4.62.1.11	<a href="#">new_set_map</a>	178



4.62.1.12	<code>new_set_nomap</code>	178
4.62.1.13	<code>peek_element</code>	178
4.62.1.14	<code>remove_element</code>	178
4.62.1.15	<code>remove_element_key</code>	178
4.62.1.16	<code>remove_element_with_key</code>	178
4.62.1.17	<code>remove_int_element</code>	178
4.62.1.18	<code>reset_set</code>	178
4.62.1.19	<code>set_empty</code>	178
4.62.1.20	<code>set_to_list</code>	178
4.63	<code>set.h</code> File Reference	178
4.63.1	Detailed Description	179
4.63.2	Typedef Documentation	179
4.63.2.1	<code>s_node</code>	179
4.63.2.2	<code>set</code>	179
4.63.3	Function Documentation	179
4.63.3.1	<code>add_element</code>	179
4.63.3.2	<code>add_element_node</code>	179
4.63.3.3	<code>add_element_nomap</code>	179
4.63.3.4	<code>add_int_element</code>	179
4.63.3.5	<code>copy_set</code>	179
4.63.3.6	<code>destroy_set</code>	179
4.63.3.7	<code>get_element_with_key</code>	179
4.63.3.8	<code>get_int_element</code>	180
4.63.3.9	<code>new_s_node</code>	180
4.63.3.10	<code>new_set</code>	180
4.63.3.11	<code>new_set_map</code>	180
4.63.3.12	<code>new_set_nomap</code>	180
4.63.3.13	<code>peek_element</code>	180
4.63.3.14	<code>remove_element</code>	180
4.63.3.15	<code>remove_element_key</code>	180
4.63.3.16	<code>remove_element_with_key</code>	180
4.63.3.17	<code>remove_int_element</code>	180
4.63.3.18	<code>reset_set</code>	180
4.63.3.19	<code>set_empty</code>	180
4.63.3.20	<code>set_to_list</code>	180
4.64	<code>solution.c</code> File Reference	180
4.64.1	Function Documentation	180
4.64.1.1	<code>assignment_has_the_solution</code>	180
4.64.1.2	<code>check_in_solution</code>	180
4.64.1.3	<code>confirm_variables</code>	180

4.64.1.4	read_solution . . . . .	180
4.64.1.5	table_has_the_solution . . . . .	180
4.65	solution.h File Reference . . . . .	181
4.65.1	Detailed Description . . . . .	181
4.65.2	Typedef Documentation . . . . .	181
4.65.2.1	solution . . . . .	181
4.65.3	Function Documentation . . . . .	181
4.65.3.1	assignment_has_the_solution . . . . .	181
4.65.3.2	check_in_solution . . . . .	181
4.65.3.3	confirm_variables . . . . .	181
4.65.3.4	read_solution . . . . .	181
4.65.3.5	table_has_the_solution . . . . .	181
4.66	sort.c File Reference . . . . .	181
4.66.1	Function Documentation . . . . .	182
4.66.1.1	buble_sort . . . . .	182
4.66.1.2	buble_sort_list . . . . .	182
4.66.1.3	buble_sort_list_2 . . . . .	182
4.66.1.4	get_rand_value . . . . .	182
4.66.1.5	is_sorted . . . . .	182
4.66.1.6	set_rand_value . . . . .	182
4.66.1.7	sort_recursive . . . . .	182
4.66.1.8	sort_relative_to_col . . . . .	182
4.66.1.9	sort_w . . . . .	182
4.66.1.10	sort_w_mcols . . . . .	182
4.66.1.11	sort_w_obj . . . . .	182
4.66.1.12	sort_w_obj_mcols . . . . .	183
4.66.1.13	swap_int . . . . .	183
4.66.1.14	swap_obj . . . . .	183
4.67	sort.h File Reference . . . . .	183
4.67.1	Detailed Description . . . . .	183
4.67.2	Function Documentation . . . . .	183
4.67.2.1	buble_sort . . . . .	183
4.67.2.2	buble_sort_list . . . . .	183
4.67.2.3	buble_sort_list_2 . . . . .	183
4.67.2.4	get_rand_value . . . . .	183
4.67.2.5	is_sorted . . . . .	183
4.67.2.6	set_rand_value . . . . .	183
4.67.2.7	sort . . . . .	184
4.67.2.8	sort_recursive . . . . .	184
4.67.2.9	sort_relative_to_col . . . . .	184

---

4.67.2.10	sort_w	184
4.67.2.11	sort_w_mcols	184
4.67.2.12	sort_w_obj	184
4.67.2.13	sort_w_obj_mcols	184
4.67.2.14	swap_int	184
4.67.2.15	swap_obj	184
4.68	tree_map.c File Reference	184
4.68.1	Function Documentation	185
4.68.1.1	compress_tree_map	185
4.68.1.2	compress_tree_map_node_list	185
4.68.1.3	compress_tree_map_rec	185
4.68.1.4	compress_tuple_list	185
4.68.1.5	destroy_tree_map	185
4.68.1.6	destroy_tree_map_arrays	185
4.68.1.7	destroy_tree_map_lists	185
4.68.1.8	fetch_tuples	185
4.68.1.9	find_node_with_id	185
4.68.1.10	find_node_with_id_compressed	185
4.68.1.11	get_tuples_for_tuple	185
4.68.1.12	get_tuples_for_tuple_compressed	185
4.68.1.13	insert_tuple_into_tree	185
4.68.1.14	insert_tuple_into_tree_payload	185
4.68.1.15	new_tree_map	185
4.68.1.16	new_tree_map_node	185
4.68.1.17	reset_tree_map	185
4.68.1.18	traverse_treemap	185
4.68.1.19	tree_map_to_htable	185
4.68.1.20	tree_map_to_htable_rec	185
4.69	tree_map.h File Reference	185
4.69.1	Detailed Description	186
4.69.2	Macro Definition Documentation	186
4.69.2.1	STAR	186
4.69.3	Typedef Documentation	186
4.69.3.1	tree_map	186
4.69.3.2	tree_map_node	186
4.69.4	Function Documentation	186
4.69.4.1	compress_tree_map	186
4.69.4.2	compress_tree_map_node_list	187
4.69.4.3	compress_tree_map_rec	187
4.69.4.4	compress_tuple_list	187

4.69.4.5	destroy_tree_map	187
4.69.4.6	destroy_tree_map_arrays	187
4.69.4.7	destroy_tree_map_lists	187
4.69.4.8	fetch_tuples	187
4.69.4.9	find_node_with_id	187
4.69.4.10	find_node_with_id_compressedd	187
4.69.4.11	get_tuples_for_tuple	187
4.69.4.12	get_tuples_for_tuple_compressed	187
4.69.4.13	insert_tuple_into_tree	187
4.69.4.14	insert_tuple_into_tree_payload	187
4.69.4.15	new_tree_map	187
4.69.4.16	new_tree_map_node	187
4.69.4.17	reset_tree_map	187
4.69.4.18	traverse_treemap	187
4.69.4.19	tree_map_to_htable	187
4.69.4.20	tree_map_to_htable_rec	187
4.70	utils.c File Reference	187
4.70.1	Macro Definition Documentation	188
4.70.1.1	PROFILE_SIZE	188
4.70.2	Function Documentation	188
4.70.2.1	catcher	188
4.70.2.2	checked_free	188
4.70.2.3	checked_malloc	188
4.70.2.4	compare_ints	188
4.70.2.5	dump_trace	189
4.70.2.6	error	189
4.70.2.7	get_double	189
4.70.2.8	get_int	189
4.70.2.9	get_profile_times	189
4.70.2.10	get_string	189
4.70.2.11	max	189
4.70.2.12	print_profile	189
4.70.2.13	print_trace	189
4.70.2.14	profile_crash_all_end	189
4.70.2.15	profile_time_end	189
4.70.2.16	profile_time_start	189
4.70.2.17	reset_vars	189
4.70.2.18	set_var	189
4.70.2.19	skip_char	189
4.70.2.20	skip_space	189

4.70.2.21	skip_to_char	189
4.70.2.22	str_cat	189
4.70.2.23	str_cat_recursive	189
4.70.2.24	str_cat_sep	189
4.70.2.25	str_copy	189
4.70.2.26	str_copy_short	189
4.70.2.27	string_equal	189
4.70.2.28	string_equal_casted	189
4.70.2.29	unvisit_var	189
4.70.2.30	visit_var	189
4.70.2.31	warn	189
4.70.3	Variable Documentation	189
4.70.3.1	profile_desc	190
4.70.3.2	profiled	190
4.70.3.3	profiled_count	190
4.71	utils.h File Reference	190
4.71.1	Detailed Description	190
4.71.2	Function Documentation	191
4.71.2.1	catcher	191
4.71.2.2	checked_free	191
4.71.2.3	checked_malloc	191
4.71.2.4	compare_ints	191
4.71.2.5	dump_trace	191
4.71.2.6	error	191
4.71.2.7	get_double	191
4.71.2.8	get_int	191
4.71.2.9	get_profile_times	191
4.71.2.10	get_string	191
4.71.2.11	max	191
4.71.2.12	print_allocs	191
4.71.2.13	print_profile	191
4.71.2.14	print_trace	191
4.71.2.15	profile_crash_all_end	191
4.71.2.16	profile_time_end	191
4.71.2.17	profile_time_start	191
4.71.2.18	reset_vars	191
4.71.2.19	set_var	191
4.71.2.20	skip_char	191
4.71.2.21	skip_space	191
4.71.2.22	skip_to_char	191

4.71.2.23	str_cat	191
4.71.2.24	str_cat_recursive	191
4.71.2.25	str_cat_sep	191
4.71.2.26	str_copy	191
4.71.2.27	str_copy_short	191
4.71.2.28	string_equal	192
4.71.2.29	string_equal_casted	192
4.71.2.30	unvisit_var	192
4.71.2.31	visit_var	192
4.71.2.32	warn	192
4.72	variables.c File Reference	192
4.72.1	Function Documentation	192
4.72.1.1	add_variable	192
4.72.1.2	compare_variable_number	192
4.72.1.3	get_variable_number	192
4.72.1.4	init_domain_sets	192
4.72.1.5	init_variables	192
4.72.1.6	new_var_val	193
4.72.1.7	no_solutions	193
4.72.1.8	print_assigned_vars	193
4.72.1.9	print_solution	193
4.72.1.10	print_variable_domain_size	193
4.72.1.11	print_variables	193
4.72.1.12	print_vars_for_sol_checker	193
4.72.1.13	reset_variable_constraints	193
4.72.1.14	scope_contained_in	193
4.72.1.15	set_static_variable_ordering	193
4.72.1.16	set_value	193
4.73	variables.h File Reference	193
4.73.1	Detailed Description	194
4.73.2	Typedef Documentation	194
4.73.2.1	var_val	194
4.73.2.2	variable	194
4.73.2.3	variables	194
4.73.3	Function Documentation	194
4.73.3.1	add_variable	194
4.73.3.2	compare_variable_number	194
4.73.3.3	get_variable_number	194
4.73.3.4	init_domain_sets	194
4.73.3.5	init_variables	194

---

4.73.3.6	init_vars	194
4.73.3.7	new_var_val	194
4.73.3.8	no_solutions	194
4.73.3.9	print_assigned_vars	194
4.73.3.10	print_solution	194
4.73.3.11	print_variable_domain_size	194
4.73.3.12	print_variables	194
4.73.3.13	print_vars_for_sol_checker	195
4.73.3.14	reset_variable_constraints	195
4.73.3.15	scope_contained_in	195
4.73.3.16	set_neighbours	195
4.73.3.17	set_static_variable_ordering	195
4.73.3.18	set_value	195

**Index****195**





# Chapter 1

## Class Index

### 1.1 Class List

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

<a href="#">active_tuple</a>	5
<a href="#">btd_data</a>	6
<a href="#">btd_engine</a>	7
<a href="#">circular_array</a>	7
<a href="#">col_card</a>	8
<a href="#">comb_cdata</a>	8
<a href="#">common_var</a>	8
<a href="#">constraint</a>	9
<a href="#">constraint_graph</a>	10
<a href="#">constraint_graph_edge</a>	11
<a href="#">constraint_graph_node</a>	11
<a href="#">constraint_tree</a>	12
<a href="#">constraint_tree_node</a>	13
<a href="#">constraints</a>	13
<a href="#">decision_tree</a>	14
<a href="#">decision_tree_system</a>	15
<a href="#">decomposed_tree</a>	15
<a href="#">decomposition_info</a>	19
<a href="#">gbtd_system</a>	20
<a href="#">hashmap</a>	21
<a href="#">heap</a>	22
<a href="#">histogram</a>	23
<a href="#">htable</a>	23
<a href="#">htv_undo_data</a>	27
<a href="#">htvar</a>	27
<a href="#">lastGAC</a>	28
<a href="#">light_stack</a>	28
<a href="#">light_stack_node</a>	29
<a href="#">llist</a>	29
<a href="#">llist_node</a>	30
<a href="#">main_structure</a>	30
<a href="#">node_graph</a>	37
<a href="#">parameter</a>	38
<a href="#">pseudo_cluster</a>	38
<a href="#">rb_tree</a>	40
<a href="#">rbt_set</a>	40
<a href="#">rs2tree_configuration</a>	41
<a href="#">rs2tree_stats</a>	41

---

s_node	44
separator_info	44
set	45
solution	45
solution_structure	46
table_map	46
tree_map	47
tree_map_node	47
tree_maps	48
tripplet	49
tuple_ids	49
undo_info	49
undo_lastGAC	50
value	50
var_undo_data	51
var_val	51
var_val_ac	52
variable	52
variables	54

# Chapter 2

## File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

ac.c	57
ac.h	58
active_tuple.c	58
active_tuple.h	59
btdallsol.c	61
btdallsol.h	63
circular_array.c	65
circular_array.h	65
combinations.c	66
combinations.h	69
constraints.c	73
constraints.h	74
decomposition.c	77
decomposition.h	82
decomposition_functions.c	88
decomposition_functions.h	89
do_db.c	90
do_db.h	91
domains.c	91
domains.h	91
gbtd.c	92
gbtd.h	94
globals.c	97
globals.h	98
hashmap.c	101
hashmap.h	102
hashtable.c	103
hashtable.h	110
hashtable_search.c	117
hashtable_search.h	118
heap.c	120
heap.h	121
histogram.c	122
histogram.h	123
llist.c	123
llist.h	126
main.c	130
minibucket.c	133

minibucket.h	134
oneall.c	134
oneall.h	135
parsers.c	135
parsers.h	136
partition.c	137
partition.h	137
query_graph.c	138
query_graph.h	142
rb_tree.c	146
rb_tree.h	148
rbt_set.c	149
rbt_set.h	150
rpwc.c	151
rpwc.h	152
rr.c	154
rr.h	155
rs2.c	155
rs2.h	160
rs2tree.c	165
rs2tree.h	170
septables.c	176
septables.h	176
set.c	177
set.h	178
solution.c	180
solution.h	181
sort.c	181
sort.h	183
tree_map.c	184
tree_map.h	185
utils.c	187
utils.h	190
variables.c	192
variables.h	193

## Chapter 3

# Class Documentation

### 3.1 active\_tuple Struct Reference

```
#include <active_tuple.h>
```

#### Public Attributes

- int [pos](#)
- int [total\\_vars](#)
- int \* [var\\_map](#)
- int \* [tuple](#)
- int \* [tuple\\_counts](#)
- int \* [vars](#)
- [rb\\_tree](#) \* [all\\_ac\\_vars](#)
- int [all\\_ac\\_vars\\_count](#)
- char \*\* [var\\_names](#)
- [tree\\_map](#) \* [tm](#)
- [tree\\_maps](#) \* [tms](#)
- [llist](#) \* [added\\_tuples](#)
- [llist](#) \* [ac\\_vars\\_from\\_tuple](#)
- [llist](#) \* [common\\_cols](#)
- struct [active\\_tuple](#) \* [next](#)
- struct [active\\_tuple](#) \* [last](#)
- int [total\\_added](#)
- int [special\\_generating\\_solutions](#)
- struct [decomposed\\_tree](#) \* [treen](#)

#### 3.1.1 Member Data Documentation

3.1.1.1 [llist\\*](#) [active\\_tuple::ac\\_vars\\_from\\_tuple](#)

3.1.1.2 [llist\\*](#) [active\\_tuple::added\\_tuples](#)

3.1.1.3 [rb\\_tree\\*](#) [active\\_tuple::all\\_ac\\_vars](#)

3.1.1.4 [int](#) [active\\_tuple::all\\_ac\\_vars\\_count](#)

3.1.1.5 [llist\\*](#) [active\\_tuple::common\\_cols](#)

- 3.1.1.6 `struct active_tuple* active_tuple::last`
- 3.1.1.7 `struct active_tuple* active_tuple::next`
- 3.1.1.8 `int active_tuple::pos`
- 3.1.1.9 `int active_tuple::special_generating_solutions`
- 3.1.1.10 `tree_map* active_tuple::tm`
- 3.1.1.11 `tree_maps* active_tuple::tms`
- 3.1.1.12 `int active_tuple::total_added`
- 3.1.1.13 `int active_tuple::total_vars`
- 3.1.1.14 `struct decomposed_tree* active_tuple::treen`
- 3.1.1.15 `int* active_tuple::tuple`
- 3.1.1.16 `int* active_tuple::tuple_counts`
- 3.1.1.17 `int* active_tuple::var_map`
- 3.1.1.18 `char** active_tuple::var_names`
- 3.1.1.19 `int* active_tuple::vars`

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

- [active\\_tuple.h](#)

## 3.2 btd\_data Struct Reference

```
#include <btdallsol.h>
```

### Public Attributes

- `llist * uninit_vars`
- `llist * init_vars`
- `active_tuple * goods`
- `active_tuple * nogoods`

### 3.2.1 Member Data Documentation

- 3.2.1.1 `active_tuple* btd_data::goods`
- 3.2.1.2 `llist* btd_data::init_vars`
- 3.2.1.3 `active_tuple* btd_data::nogoods`
- 3.2.1.4 `llist* btd_data::uninit_vars`

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

- [btdallsol.h](#)

## 3.3 btd\_engine Struct Reference

```
#include <btdallsol.h>
```

### Public Attributes

- [decomposed\\_tree](#) \* [tree\\_root](#)
- [decomposed\\_tree](#) \* [current\\_cluster](#)
- [llist\\_node](#) \* [next\\_variable\\_in\\_current\\_cluster\\_to\\_select](#)
- [decomposed\\_tree](#) \*\* [variable\\_highest\\_cluster](#)

### 3.3.1 Member Data Documentation

3.3.1.1 [decomposed\\_tree](#)\* [btd\\_engine::current\\_cluster](#)

3.3.1.2 [llist\\_node](#)\* [btd\\_engine::next\\_variable\\_in\\_current\\_cluster\\_to\\_select](#)

3.3.1.3 [decomposed\\_tree](#)\* [btd\\_engine::tree\\_root](#)

3.3.1.4 [decomposed\\_tree](#)\*\* [btd\\_engine::variable\\_highest\\_cluster](#)

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

- [btdallsol.h](#)

## 3.4 circular\_array Struct Reference

```
#include <circular_array.h>
```

### Public Attributes

- int [i](#)
- int [size](#)
- int \* [array](#)

### 3.4.1 Member Data Documentation

3.4.1.1 int\* [circular\\_array::array](#)

3.4.1.2 int [circular\\_array::i](#)

3.4.1.3 int [circular\\_array::size](#)

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

- [circular\\_array.h](#)

## 3.5 col\_card Struct Reference

```
#include <hashtable.h>
```

### Public Attributes

- char \* [col\\_name](#)
- int [card](#)

### 3.5.1 Member Data Documentation

3.5.1.1 int col\_card::card

3.5.1.2 char\* col\_card::col\_name

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

- [hashtable.h](#)

## 3.6 comb\_cdata Struct Reference

```
#include <rs2.h>
```

### Public Attributes

- [llist](#) \* [comb](#)
- [comb\\_data](#) \* [cd](#)
- int [context](#)

### 3.6.1 Member Data Documentation

3.6.1.1 [comb\\_data](#)\* [comb\\_cdata](#)::[cd](#)

3.6.1.2 [llist](#)\* [comb\\_cdata](#)::[comb](#)

3.6.1.3 int [comb\\_cdata](#)::[context](#)

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

- [rs2.h](#)

## 3.7 common\_var Struct Reference

```
#include <hashtable.h>
```

### Public Attributes

- int [apos](#)
- int [bpos](#)



### 3.7.1 Member Data Documentation

3.7.1.1 `int common_var::apos`

3.7.1.2 `int common_var::bpos`

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

- [hashtable.h](#)

## 3.8 constraint Struct Reference

```
#include <constraints.h>
```

### Public Attributes

- `char * name`
- `int function`
- `int constraint_id`
- `parameter ** parameters`
- `tuple * tuples`
- `int * variables`
- `llist_node *** ac3_1_last`
- `int uninit_vars`
- `llist * qgn_references`
- `char decomposition_added`
- `struct htable * ht`
- `int fill`
- `int lastdftid`

### 3.8.1 Member Data Documentation

3.8.1.1 `llist_node*** constraint::ac3_1_last`

3.8.1.2 `int constraint::constraint_id`

3.8.1.3 `char constraint::decomposition_added`

3.8.1.4 `int constraint::fill`

3.8.1.5 `int constraint::function`

3.8.1.6 `struct htable* constraint::ht`

3.8.1.7 `int constraint::lastdftid`

3.8.1.8 `char* constraint::name`

3.8.1.9 `parameter** constraint::parameters`

3.8.1.10 `llist* constraint::qgn_references`

3.8.1.11 `tuple* constraint::tuples`

3.8.1.12 `int constraint::uninit_vars`

3.8.1.13 `int* constraint::variables`

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

- [constraints.h](#)

## 3.9 `constraint_graph` Struct Reference

```
#include <query_graph.h>
```

### Public Attributes

- `constraint * root`
- `int constraint_node_count`
- `int edge_count`
- `int cgn_i_size`
- `constraint_graph_node ** cgn_i`
- `llist * node_list`
- `constraint_graph_edge ** cge_i`
- `llist * edge_list`
- `constraint_graph_edge *** matrix`
- `llist * left_deep_list`
- `int fill_edges_count`
- `int max_arity`

### 3.9.1 Member Data Documentation

3.9.1.1 `constraint_graph_edge** constraint_graph::cge_i`

3.9.1.2 `constraint_graph_node** constraint_graph::cgn_i`

3.9.1.3 `int constraint_graph::cgn_i_size`

3.9.1.4 `int constraint_graph::constraint_node_count`

3.9.1.5 `int constraint_graph::edge_count`

3.9.1.6 `llist* constraint_graph::edge_list`

3.9.1.7 `int constraint_graph::fill_edges_count`

3.9.1.8 `llist* constraint_graph::left_deep_list`

3.9.1.9 `constraint_graph_edge*** constraint_graph::matrix`

3.9.1.10 `int constraint_graph::max_arity`

3.9.1.11 `llist* constraint_graph::node_list`

3.9.1.12 `constraint* constraint_graph::root`

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

- [query\\_graph.h](#)

## 3.10 constraint\_graph\_edge Struct Reference

```
#include <query_graph.h>
```

### Public Attributes

- int [id](#)
- [constraint\\_graph\\_node](#) \* [node1](#)
- [constraint\\_graph\\_node](#) \* [node2](#)
- int [weight](#)
- [htable](#) \* [table](#)
- [q\\_node](#) \* [query\\_node](#)
- [llist](#) \* [common\\_vars](#)

### 3.10.1 Member Data Documentation

3.10.1.1 [llist](#)\* [constraint\\_graph\\_edge::common\\_vars](#)

3.10.1.2 int [constraint\\_graph\\_edge::id](#)

3.10.1.3 [constraint\\_graph\\_node](#)\* [constraint\\_graph\\_edge::node1](#)

3.10.1.4 [constraint\\_graph\\_node](#)\* [constraint\\_graph\\_edge::node2](#)

3.10.1.5 [q\\_node](#)\* [constraint\\_graph\\_edge::query\\_node](#)

3.10.1.6 [htable](#)\* [constraint\\_graph\\_edge::table](#)

3.10.1.7 int [constraint\\_graph\\_edge::weight](#)

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

- [query\\_graph.h](#)

## 3.11 constraint\_graph\_node Struct Reference

```
#include <query_graph.h>
```

### Public Attributes

- [constraint](#) \* [constr](#)
- [rbt\\_set](#) \* [neighbours](#)
- int [color](#)
- char \* [name](#)
- [htable](#) \* [hash\\_table](#)
- int [table\\_changed\\_after\\_suspend](#)
- int [id](#)
- int [generation](#)
- int [pgeneration](#)

- int [combinations\\_removed](#)
- [llist](#) \* children
- [llist](#) \* parents
- int [out\\_of\\_order](#)
- int [peo](#)
- int [fill\\_in](#)
- int [heap\\_pos](#)

### 3.11.1 Member Data Documentation

3.11.1.1 [llist](#)\* [constraint\\_graph\\_node::children](#)

3.11.1.2 int [constraint\\_graph\\_node::color](#)

3.11.1.3 int [constraint\\_graph\\_node::combinations\\_removed](#)

3.11.1.4 [constraint](#)\* [constraint\\_graph\\_node::constr](#)

3.11.1.5 int [constraint\\_graph\\_node::fill\\_in](#)

3.11.1.6 int [constraint\\_graph\\_node::generation](#)

3.11.1.7 [htable](#)\* [constraint\\_graph\\_node::hash\\_table](#)

3.11.1.8 int [constraint\\_graph\\_node::heap\\_pos](#)

3.11.1.9 int [constraint\\_graph\\_node::id](#)

3.11.1.10 char\* [constraint\\_graph\\_node::name](#)

3.11.1.11 [rbt\\_set](#)\* [constraint\\_graph\\_node::neighbours](#)

3.11.1.12 int [constraint\\_graph\\_node::out\\_of\\_order](#)

3.11.1.13 [llist](#)\* [constraint\\_graph\\_node::parents](#)

3.11.1.14 int [constraint\\_graph\\_node::peo](#)

3.11.1.15 int [constraint\\_graph\\_node::pgeneration](#)

3.11.1.16 int [constraint\\_graph\\_node::table\\_changed\\_after\\_suspend](#)

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

- [query\\_graph.h](#)

## 3.12 [constraint\\_tree](#) Struct Reference

```
#include <constraints.h>
```

### Public Attributes

- int [id](#)

- struct [constraint\\_tree\\_node](#) \* [root](#)
- int [nodes](#)
- int [width](#)
- struct [constraint\\_tree\\_node](#) \*\* [clusters](#)

### 3.12.1 Member Data Documentation

3.12.1.1 struct [constraint\\_tree\\_node](#)\*\* [constraint\\_tree::clusters](#)

3.12.1.2 int [constraint\\_tree::id](#)

3.12.1.3 int [constraint\\_tree::nodes](#)

3.12.1.4 struct [constraint\\_tree\\_node](#)\* [constraint\\_tree::root](#)

3.12.1.5 int [constraint\\_tree::width](#)

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

- [constraints.h](#)

## 3.13 constraint\_tree\_node Struct Reference

```
#include <constraints.h>
```

### Public Attributes

- int [id](#)
- [llist](#) \* [htables](#)
- struct [constraint\\_tree\\_node](#) \* [parent](#)
- [llist](#) \* [children](#)

### 3.13.1 Member Data Documentation

3.13.1.1 [llist](#)\* [constraint\\_tree\\_node::children](#)

3.13.1.2 [llist](#)\* [constraint\\_tree\\_node::htables](#)

3.13.1.3 int [constraint\\_tree\\_node::id](#)

3.13.1.4 struct [constraint\\_tree\\_node](#)\* [constraint\\_tree\\_node::parent](#)

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

- [constraints.h](#)

## 3.14 constraints Struct Reference

```
#include <constraints.h>
```

## Public Attributes

- int [size](#)
- int [count](#)
- [constraint](#) \* [constraint\\_list](#)
- int [max\\_arity](#)
- int [has\\_intension](#)
- struct [rb\\_tree](#) \* [tables](#)
- int [last\\_id](#)

### 3.14.1 Member Data Documentation

3.14.1.1 [constraint](#)\* [constraints::constraint\\_list](#)

3.14.1.2 int [constraints::count](#)

3.14.1.3 int [constraints::has\\_intension](#)

3.14.1.4 int [constraints::last\\_id](#)

3.14.1.5 int [constraints::max\\_arity](#)

3.14.1.6 int [constraints::size](#)

3.14.1.7 struct [rb\\_tree](#)\* [constraints::tables](#)

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

- [constraints.h](#)

## 3.15 [decision\\_tree](#) Struct Reference

```
#include <oneall.h>
```

### Public Attributes

- int [id](#)
- double [value](#)
- int [level](#)
- char \* [attribute](#)
- struct [decision\\_tree](#) \* [left](#)
- struct [decision\\_tree](#) \* [right](#)
- struct [decision\\_tree](#) \* [parent](#)

### 3.15.1 Member Data Documentation

3.15.1.1 char\* [decision\\_tree::attribute](#)

3.15.1.2 int [decision\\_tree::id](#)

3.15.1.3 struct [decision\\_tree](#)\* [decision\\_tree::left](#)

3.15.1.4 `int decision_tree::level`

3.15.1.5 `struct decision_tree* decision_tree::parent`

3.15.1.6 `struct decision_tree* decision_tree::right`

3.15.1.7 `double decision_tree::value`

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

- [oneall.h](#)

## 3.16 `decision_tree_system` Struct Reference

```
#include <oneall.h>
```

### Public Attributes

- `char ** names`
- `decision_tree * dt`

### 3.16.1 Member Data Documentation

3.16.1.1 `decision_tree* decision_tree_system::dt`

3.16.1.2 `char** decision_tree_system::names`

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

- [oneall.h](#)

## 3.17 `decomposed_tree` Struct Reference

```
#include <decomposition.h>
```

### Public Attributes

- `int id`
- `int clique_number`
- `int dfid`
- `llist * children`
- `struct decomposed_tree * parent`
- `set * vars`
- `llist * varlist`
- `set * relation_vars`
- `set * separator_vars`
- `llist * constraints`
- `llist * projected_relations`
- `llist * comb_datas`
- `llist * combs`
- `llist * separator`

- `llist * separator_var_names`
- `llist * cluster_var_names`
- `llist * separator_rels`
- `llist * projected_rels_on_c_vars`
- `llist * context_rels`
- `llist * projected_rels_on_sep_vars`
- `llist * mini_bucket_rels`
- `llist * uncovered_vars`
- `llist * affiliate_rels`
- `llist ** affiliates_map`
- `llist * filter_more`
- `llist * filter_cluster_relation`
- `llist * filter_cluster_relation_reverse`
- `int pre`
- `int post`
- `tree_map * parent_separator_tm`
- `active_tuple * parent_separator_ac`
- `llist * parent_separator_tm_common_cols`
- `htable * ancestor_table`
- `llist * cgs`
- `set * queue`
- `llist * fill_edge_constraints`
- `int level`
- `int depth`
- `htable * cluster_relation`
- `htable * separator_relation`
- `llist * separator_relations`
- `struct btd_data * btddata`
- `llist_node * position_in_parent_node_children_list`
- `llist * combinations`
- `int subtree_size`
- `rb_tree * tables`
- `rb_tree * f_tables`
- `rb_tree * ignore_tables`
- `comb_data ** representative_comb_datas`
- `llist * glue_combs`
- `double weight`
- `int children_not_processed`
- `int heap_pos`
- `int degree`
- `struct decomposed_tree ** adjacency`
- `separator_info * separators`
- `llist * leafs`
- `int visit_stamp`
- `int in_fringe`
- `rb_tree * stats`
- `rb_tree * feature_stats`
- `struct decomposed_tree ** c_array`
- `llist * separator_scopes`
- `int has_stats`
- `int pending_sep_table_generation`



### 3.17.1 Member Data Documentation

3.17.1.1 struct decomposed\_tree\*\* decomposed\_tree::adjacency

3.17.1.2 llist\* decomposed\_tree::affiliate\_rels

3.17.1.3 llist\*\* decomposed\_tree::affiliates\_map

3.17.1.4 htable\* decomposed\_tree::ancestor\_table

3.17.1.5 struct btd\_data\* decomposed\_tree::btddata

3.17.1.6 struct decomposed\_tree\*\* decomposed\_tree::c\_array

3.17.1.7 llist\* decomposed\_tree::cgs

3.17.1.8 llist\* decomposed\_tree::children

3.17.1.9 int decomposed\_tree::children\_not\_processed

3.17.1.10 int decomposed\_tree::clique\_number

3.17.1.11 htable\* decomposed\_tree::cluster\_relation

3.17.1.12 llist\* decomposed\_tree::cluster\_var\_names

3.17.1.13 llist\* decomposed\_tree::comb\_datas

3.17.1.14 llist\* decomposed\_tree::combinations

3.17.1.15 llist\* decomposed\_tree::combs

3.17.1.16 llist\* decomposed\_tree::constraints

3.17.1.17 llist\* decomposed\_tree::context\_rels

3.17.1.18 int decomposed\_tree::degree

3.17.1.19 int decomposed\_tree::depth

3.17.1.20 int decomposed\_tree::dftid

3.17.1.21 rb\_tree\* decomposed\_tree::f\_tables

3.17.1.22 rb\_tree\* decomposed\_tree::feature\_stats

3.17.1.23 llist\* decomposed\_tree::fill\_edge\_constraints

3.17.1.24 llist\* decomposed\_tree::filter\_cluster\_relation

3.17.1.25 llist\* decomposed\_tree::filter\_cluster\_relation\_reverse

3.17.1.26 llist\* decomposed\_tree::filter\_more

3.17.1.27 llist\* decomposed\_tree::glue\_combs

- 3.17.1.28 `int decomposed_tree::has_stats`
- 3.17.1.29 `int decomposed_tree::heap_pos`
- 3.17.1.30 `int decomposed_tree::id`
- 3.17.1.31 `rb_tree* decomposed_tree::ignore_tables`
- 3.17.1.32 `int decomposed_tree::in_fringe`
- 3.17.1.33 `llist* decomposed_tree::leafs`
- 3.17.1.34 `int decomposed_tree::level`
- 3.17.1.35 `llist* decomposed_tree::mini_bucket_rels`
- 3.17.1.36 `struct decomposed_tree* decomposed_tree::parent`
- 3.17.1.37 `active_tuple* decomposed_tree::parent_separator_ac`
- 3.17.1.38 `tree_map* decomposed_tree::parent_separator_tm`
- 3.17.1.39 `llist* decomposed_tree::parent_separator_tm_common_cols`
- 3.17.1.40 `int decomposed_tree::pending_sep_table_generation`
- 3.17.1.41 `llist_node* decomposed_tree::position_in_parent_node_children_list`
- 3.17.1.42 `int decomposed_tree::post`
- 3.17.1.43 `int decomposed_tree::pre`
- 3.17.1.44 `llist* decomposed_tree::projected_relations`
- 3.17.1.45 `llist* decomposed_tree::projected_rels_on_c_vars`
- 3.17.1.46 `llist* decomposed_tree::projected_rels_on_sep_vars`
- 3.17.1.47 `set* decomposed_tree::queue`
- 3.17.1.48 `set* decomposed_tree::relation_vars`
- 3.17.1.49 `comb_data** decomposed_tree::representative_comb_datas`
- 3.17.1.50 `llist* decomposed_tree::separator`
- 3.17.1.51 `htable* decomposed_tree::separator_relation`
- 3.17.1.52 `llist* decomposed_tree::separator_relations`
- 3.17.1.53 `llist* decomposed_tree::separator_rels`
- 3.17.1.54 `llist* decomposed_tree::separator_scopes`
- 3.17.1.55 `llist* decomposed_tree::separator_var_names`

- 3.17.1.56 `set*` `decomposed_tree::separator_vars`
- 3.17.1.57 `separator_info*` `decomposed_tree::separators`
- 3.17.1.58 `rb_tree*` `decomposed_tree::stats`
- 3.17.1.59 `int` `decomposed_tree::subtree_size`
- 3.17.1.60 `rb_tree*` `decomposed_tree::tables`
- 3.17.1.61 `llist*` `decomposed_tree::uncovered_vars`
- 3.17.1.62 `llist*` `decomposed_tree::varlist`
- 3.17.1.63 `set*` `decomposed_tree::vars`
- 3.17.1.64 `int` `decomposed_tree::visit_stamp`
- 3.17.1.65 `double` `decomposed_tree::weight`

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

- [decomposition.h](#)

## 3.18 decomposition\_info Struct Reference

```
#include <decomposition.h>
```

### Public Attributes

- `int` [number\\_of\\_clusters](#)
- `int` [treewidth](#)
- `int` [hyperwidth](#)
- `int` [min\\_hyperwidth](#)
- `int` [largest\\_separator\\_size](#)
- `int` [largest\\_separator\\_size\\_rels](#)
- `int` [mtreewidth](#)
- `int` [mhyperwidth](#)
- `int` [smallest\\_separator\\_size](#)
- `int` [smallest\\_separator\\_size\\_rels](#)

### 3.18.1 Member Data Documentation

- 3.18.1.1 `int` `decomposition_info::hyperwidth`
- 3.18.1.2 `int` `decomposition_info::largest_separator_size`
- 3.18.1.3 `int` `decomposition_info::largest_separator_size_rels`
- 3.18.1.4 `int` `decomposition_info::mhyperwidth`
- 3.18.1.5 `int` `decomposition_info::min_hyperwidth`

3.18.1.6 int decomposition\_info::mtreewidth

3.18.1.7 int decomposition\_info::number\_of\_clusters

3.18.1.8 int decomposition\_info::smallest\_separator\_size

3.18.1.9 int decomposition\_info::smallest\_separator\_size\_rels

3.18.1.10 int decomposition\_info::treewidth

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

- [decomposition.h](#)

## 3.19 gbtd\_system Struct Reference

```
#include <gbtd.h>
```

### Public Attributes

- [pseudo\\_cluster](#) \* root
- [pseudo\\_cluster](#) \* current\_cluster
- [llist](#) \* vars
- void \* [current\\_var](#)
- [options](#) system
- double [solCount](#)
- int [isolCount](#)
- int [total\\_vars](#)
- int [unassigned\\_vars](#)
- int [total\\_clusters](#)
- int [allSol](#)
- [set](#) \*\* [queues](#)
- [set](#) \* [queue](#)
- [decomposed\\_tree](#) \*\* [treens](#)
- [llist\\_node](#) \* [current\\_var\\_p](#)
- int [count\\_on\\_m\\_s](#)
- [active\\_tuple](#) \* [ac](#)
- [comb\\_data](#) \* [cd](#)
- int [revision\\_time](#)
- int \* [tuple\\_positions](#)
- int \* [tuple\\_position\\_htid](#)
- int [wtn](#)

### 3.19.1 Member Data Documentation

3.19.1.1 [active\\_tuple](#)\* gbtd\_system::ac

3.19.1.2 int gbtd\_system::allSol

3.19.1.3 [comb\\_data](#)\* gbtd\_system::cd

3.19.1.4 int gbtd\_system::count\_on\_m\_s

- 3.19.1.5 `pseudo_cluster*` `gbtd_system::current_cluster`
- 3.19.1.6 `void*` `gbtd_system::current_var`
- 3.19.1.7 `llist_node*` `gbtd_system::current_var_p`
- 3.19.1.8 `int` `gbtd_system::isolCount`
- 3.19.1.9 `set*` `gbtd_system::queue`
- 3.19.1.10 `set**` `gbtd_system::queues`
- 3.19.1.11 `int` `gbtd_system::revision_time`
- 3.19.1.12 `pseudo_cluster*` `gbtd_system::root`
- 3.19.1.13 `double` `gbtd_system::solCount`
- 3.19.1.14 `options` `gbtd_system::system`
- 3.19.1.15 `int` `gbtd_system::total_clusters`
- 3.19.1.16 `int` `gbtd_system::total_vars`
- 3.19.1.17 `decomposed_tree**` `gbtd_system::treens`
- 3.19.1.18 `int*` `gbtd_system::tuple_position_htid`
- 3.19.1.19 `int*` `gbtd_system::tuple_positions`
- 3.19.1.20 `int` `gbtd_system::unassigned_vars`
- 3.19.1.21 `llist*` `gbtd_system::vars`
- 3.19.1.22 `int` `gbtd_system::wtn`

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

- [gbtd.h](#)

## 3.20 hashmap Struct Reference

```
#include <hashmap.h>
```

### Public Attributes

- `int` `id`
- `int` `size`
- `int` `longest_chain`
- `int` `inserted`
- `llist **` `datap`
- `int` `width`
- `int *` `col_pos`
- `int *` `max`

- [KEYTYPE](#) \* [hashmap](#)
- int [single](#)
- struct [htable](#) \* [ht](#)
- char \* [note](#)
- [llist](#) \* [iterator](#)
- [llist\\_node](#) \* [iteratorptr](#)

### 3.20.1 Member Data Documentation

3.20.1.1 int\* [hashmap::col\\_pos](#)

3.20.1.2 [llist](#)\*\* [hashmap::datap](#)

3.20.1.3 [KEYTYPE](#)\* [hashmap::hashmap](#)

3.20.1.4 struct [htable](#)\* [hashmap::ht](#)

3.20.1.5 int [hashmap::id](#)

3.20.1.6 int [hashmap::inserted](#)

3.20.1.7 [llist](#)\* [hashmap::iterator](#)

3.20.1.8 [llist\\_node](#)\* [hashmap::iteratorptr](#)

3.20.1.9 int [hashmap::longest\\_chain](#)

3.20.1.10 int\* [hashmap::max](#)

3.20.1.11 char\* [hashmap::note](#)

3.20.1.12 int [hashmap::single](#)

3.20.1.13 int [hashmap::size](#)

3.20.1.14 int [hashmap::width](#)

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

- [hashmap.h](#)

## 3.21 heap Struct Reference

```
#include <heap.h>
```

### Public Attributes

- void \*\* [body](#)
- int [size](#)
- int [count](#)

### 3.21.1 Member Data Documentation

3.21.1.1 `void** heap::body`

3.21.1.2 `int heap::count`

3.21.1.3 `int heap::size`

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

- [heap.h](#)

## 3.22 histogram Struct Reference

```
#include <hashtable.h>
```

### Public Attributes

- `int columns`
- `int * values`
- `int * min`
- `int * max`
- `int * dist`
- `int ** histo`

### 3.22.1 Member Data Documentation

3.22.1.1 `int histogram::columns`

3.22.1.2 `int* histogram::dist`

3.22.1.3 `int** histogram::histo`

3.22.1.4 `int* histogram::max`

3.22.1.5 `int* histogram::min`

3.22.1.6 `int* histogram::values`

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

- [hashtable.h](#)

## 3.23htable Struct Reference

```
#include <hashtable.h>
```

### Public Attributes

- `int capacity`
- `int total_hight`

- int `last_pos`
- int `width`
- int `id`
- int `tmpid`
- char \* `name`
- char \* `short_name`
- int `is_first`
- int `join_cutoff`
- char \*\* `col_names`
- llist \* `col_names_l`
- int \* `vars`
- int `clustered1`
- int `clustered2`
- int \*\* `table`
- llist \* `tuple_list`
- llist\_node \*\* `tuple_list_array`
- llist \*\* `tuple_pointed_at_from`
- llist \*\* `supports`
- llist\_node \*\* `deleted_time_node`
- llist \* `deleted_times`
- struct `htable` \* `next`
- struct `htable` \* `prev`
- struct `htable` \* `last`
- struct `histogram` \* `histo`
- double \* `entropy`
- llist \* `tree_maps`
- int `complete`
- int `incomplete_i`
- int `incomplete_j`
- int `incomplete_clustereda`
- int `incomplete_clusteredb`
- int `in_tree`
- int `no_destroy`
- `constraint` \* `constr`
- llist \* `in_combinations`
- llist \* `in_combinations_hashed`
- llist \* `index_of_ht_in_combinations`
- int `deleted`
- int `iterator`
- `hashmap` \* `htiterator`
- int \* `com_cols_with_next_in_order`
- struct `tree_node` \* `tn`
- struct `constraint_graph_node` \* `cgn`
- `hashmap` \* `jthm`
- struct `htable` \* `jthmht`
- int \* `jcolpos`
- llist \*\* `neighbour_list_per_tuple`
- int `context`
- llist \* `comb_datas`
- int `tuple_lost`
- double \* `tuple_values`
- int `problemtable`
- int `flag`
- int `markgeneration`
- `options` type
- int `assigned`



### 3.23.1 Member Data Documentation

- 3.23.1.1 `int htable::assigned`
- 3.23.1.2 `int htable::capacity`
- 3.23.1.3 `struct constraint_graph_node* htable::cgn`
- 3.23.1.4 `int htable::clustered1`
- 3.23.1.5 `int htable::clustered2`
- 3.23.1.6 `char** htable::col_names`
- 3.23.1.7 `llist* htable::col_names_l`
- 3.23.1.8 `int* htable::com_cols_with_next_in_order`
- 3.23.1.9 `llist* htable::comb_datas`
- 3.23.1.10 `int htable::complete`
- 3.23.1.11 `constraint* htable::constr`
- 3.23.1.12 `int htable::context`
- 3.23.1.13 `int htable::deleted`
- 3.23.1.14 `llist_node** htable::deleted_time_node`
- 3.23.1.15 `llist* htable::deleted_times`
- 3.23.1.16 `double* htable::entropy`
- 3.23.1.17 `int htable::flag`
- 3.23.1.18 `struct histogram* htable::histo`
- 3.23.1.19 `hashmap* htable::htiterator`
- 3.23.1.20 `int htable::id`
- 3.23.1.21 `llist* htable::in_combinations`
- 3.23.1.22 `llist* htable::in_combinations_hashed`
- 3.23.1.23 `int htable::in_tree`
- 3.23.1.24 `int htable::incomplete_clustereda`
- 3.23.1.25 `int htable::incomplete_clusteredb`
- 3.23.1.26 `int htable::incomplete_i`
- 3.23.1.27 `int htable::incomplete_j`

- 3.23.1.28 **llist\*** `htable::index_of_ht_in_combinations`
- 3.23.1.29 **int** `htable::is_first`
- 3.23.1.30 **int** `htable::iterator`
- 3.23.1.31 **int** `htable::join_cutoff`
- 3.23.1.32 **int\*** `htable::jtcolpos`
- 3.23.1.33 **hashmap\*** `htable::jthm`
- 3.23.1.34 **struct** `htable*` `htable::jthmht`
- 3.23.1.35 **struct** `htable*` `htable::last`
- 3.23.1.36 **int** `htable::last_pos`
- 3.23.1.37 **int** `htable::markgeneration`
- 3.23.1.38 **char\*** `htable::name`
- 3.23.1.39 **llist\*\*** `htable::neighbour_list_per_tuple`
- 3.23.1.40 **struct** `htable*` `htable::next`
- 3.23.1.41 **int** `htable::no_destroy`
- 3.23.1.42 **struct** `htable*` `htable::prev`
- 3.23.1.43 **int** `htable::problemtable`
- 3.23.1.44 **char\*** `htable::short_name`
- 3.23.1.45 **llist\*\*** `htable::supports`
- 3.23.1.46 **int\*\*** `htable::table`
- 3.23.1.47 **int** `htable::tmpid`
- 3.23.1.48 **struct** `tree_node*` `htable::tn`
- 3.23.1.49 **int** `htable::total_hight`
- 3.23.1.50 **llist\*** `htable::tree_maps`
- 3.23.1.51 **llist\*** `htable::tuple_list`
- 3.23.1.52 **llist\_node\*\*** `htable::tuple_list_array`
- 3.23.1.53 **int** `htable::tuple_lost`
- 3.23.1.54 **llist\*\*** `htable::tuple_pointed_at_from`
- 3.23.1.55 **double\*** `htable::tuple_values`

3.23.1.56 options htable::type

3.23.1.57 int\* htable::vars

3.23.1.58 int htable::width

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

- [hashtable.h](#)

## 3.24 htv\_undo\_data Struct Reference

```
#include <gbtd.h>
```

### Public Attributes

- [htvar \\* htv](#)
- [llist \\* undo\\_deg](#)
- [int \\* min\\_marks](#)
- [llist \\* list\\_of\\_htvar\\_toundo\\_domlist\\_stack](#)

### 3.24.1 Member Data Documentation

3.24.1.1 [htvar\\* htv\\_undo\\_data::htv](#)

3.24.1.2 [llist\\* htv\\_undo\\_data::list\\_of\\_htvar\\_toundo\\_domlist\\_stack](#)

3.24.1.3 [int\\* htv\\_undo\\_data::min\\_marks](#)

3.24.1.4 [llist\\* htv\\_undo\\_data::undo\\_deg](#)

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

- [gbtd.h](#)

## 3.25 htvar Struct Reference

```
#include <gbtd.h>
```

### Public Attributes

- [htable \\* ht](#)
- [int vid](#)
- [int \\* chosen\\_value](#)
- [int value\\_lock](#)
- [char current\\_domain\\_tm\\_domlist](#)
- [llist\\_node \\* cd\\_cursor](#)
- [int tm\\_cursor](#)
- [llist \\* domlist\\_stack](#)
- [int dom\\_count](#)

### 3.25.1 Member Data Documentation

3.25.1.1 `llist_node*` `htvar::cd_cursor`

3.25.1.2 `int*` `htvar::chosen_value`

3.25.1.3 `char` `htvar::current_domain_tm_domlist`

3.25.1.4 `int` `htvar::dom_count`

3.25.1.5 `llist*` `htvar::domlist_stack`

3.25.1.6 `htable*` `htvar::ht`

3.25.1.7 `int` `htvar::tm_cursor`

3.25.1.8 `int` `htvar::value_lock`

3.25.1.9 `int` `htvar::vid`

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

- [gbtd.h](#)

### 3.26 lastGAC Struct Reference

```
#include <rpwc.h>
```

#### Public Attributes

- `htable *` `ht`
- `int` `pos`
- `int` `vpos`

### 3.26.1 Member Data Documentation

3.26.1.1 `htable*` `lastGAC::ht`

3.26.1.2 `int` `lastGAC::pos`

3.26.1.3 `int` `lastGAC::vpos`

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

- [rpwc.h](#)

### 3.27 light\_stack Struct Reference

```
#include <llist.h>
```

## Public Attributes

- struct [light\\_stack\\_node](#) \* `head`

### 3.27.1 Member Data Documentation

#### 3.27.1.1 struct `light_stack_node`\* `light_stack::head`

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

- [llist.h](#)

## 3.28 `light_stack_node` Struct Reference

```
#include <llist.h>
```

## Public Attributes

- int `value`
- void \* `body`
- struct [light\\_stack\\_node](#) \* `next`

### 3.28.1 Member Data Documentation

#### 3.28.1.1 void\* `light_stack_node::body`

#### 3.28.1.2 struct `light_stack_node`\* `light_stack_node::next`

#### 3.28.1.3 int `light_stack_node::value`

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

- [llist.h](#)

## 3.29 `llist` Struct Reference

```
#include <llist.h>
```

## Public Attributes

- struct [llist\\_node](#) \* `head`
- struct [llist\\_node](#) \* `tail`
- struct [llist\\_node](#) \* `min`
- struct [llist\\_node](#) \* `max`
- int `id`
- int `count`

### 3.29.1 Member Data Documentation

3.29.1.1 `int llist::count`

3.29.1.2 `struct llist_node* llist::head`

3.29.1.3 `int llist::id`

3.29.1.4 `struct llist_node* llist::max`

3.29.1.5 `struct llist_node* llist::min`

3.29.1.6 `struct llist_node* llist::tail`

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

- [llist.h](#)

## 3.30 llist\_node Struct Reference

```
#include <llist.h>
```

### Public Attributes

- `void * body`
- `int ibody`
- `struct llist_node * next`
- `struct llist_node * previous`
- `struct llist * list`

### 3.30.1 Member Data Documentation

3.30.1.1 `void* llist_node::body`

3.30.1.2 `int llist_node::ibody`

3.30.1.3 `struct llist* llist_node::list`

3.30.1.4 `struct llist_node* llist_node::next`

3.30.1.5 `struct llist_node* llist_node::previous`

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

- [llist.h](#)

## 3.31 main\_structure Struct Reference

```
#include <globals.h>
```

## Public Attributes

- int `valid`
- int `partition_id`
- int `function_offset`
- `variables * vars`
- `constraints * cons`
- `llist * domains`
- int \* `expressions`
- `relation * relations`
- int `expression_pointer`
- int `check_counter`
- int `visit_counter`
- int `bt_visit_counter`
- int `bt_counter`
- int `solve`
- int `all_solutions`
- int `init_count`
- int `consistant_solution_count`
- int `consistant_print`
- int `variable_choice_huristic`
- int `max_domain_value`
- int `v`
- int \*\* `temp`
- int `temp_count`
- int `fc_cbj_fc`
- int `ac3`
- options `tree_decom`
- options `tree_decom_solve`
- options `tree_decom_solve_1_all`
- int `tree_decom_wb`
- int `apply_td`
- `alive_table * at`
- void \* `tempp`
- int `left_deep`
- int `project`
- int `forward_check`
- int `use_histo`
- int `construct_solution`
- clock\_t `split_time`
- int `only_check_connected`
- int `total_nodes`
- int `remaining_nodes`
- int `db_algo`
- double `startt`
- options `r_star_algo`
- options `r_star_all_single_sol`
- options `r_star_gac`
- int `comb_size`
- int `comb_size_search`
- int `pprs`
- int `print_chk`
- struct `alive_table * alive_t`
- struct `constraint_graph * cg`
- int \* `vmap`

- `llist * vmap_dirty_list`
- `int * cmap`
- `int rpwc`
- `int nosingleedge`
- `llist * effected_vars_by_gac`
- `int profile`
- `int m`
- `int h`
- `int r`
- `set * updated_combinations`
- `int with_index_delete Updating`
- `int with_tuple_support_recording`
- `llist * to_queue_supports`
- `int just_combinations`
- `int just_redundancy_removal`
- `int no_tuples`
- `options no_check_insol_tuples`
- `struct comb_data * delme`
- `int remove_redundant_edges`
- `int * temp_int_p`
- `char * filename`
- `int no_support`
- `int bayesian`
- `char * tempstr`
- `int split_cutoff`
- `int long_tuple_index`
- `int minibucket`
- `int minibucketopts`
- `struct solution * sol`
- `char * problemname`
- `int trace_trigger`
- `long double number_of_solutions`
- `int comb_ids`
- `int ht_id`
- `int tree_bt_time`
- `options rs2tree_scheme`
- `options rs2tree_scheme_options`
- `int rs2tree_scheme_options_parameter`
- `int feature_vector_options`
- `llist * active_htables`
- `llist * tms_list`
- `int tms_total_inserted`
- `int dcid`
- `struct main_structure * next`
- `struct main_structure * previous`
- `llist * profile_times`
- `int constraint_placement`
- `struct decomposition_info * di`
- `options queue`
- `int rel_revise_counter`
- `options search`
- `struct decision_tree_system * dts`
- `int auto_all_revise_count`
- `int auto_one_revise_count`
- `int auto_one_ht_revise_count`



- int [auto\\_one\\_ht\\_revise\\_count\\_k](#)
- int [auto\\_all\\_ht\\_revise\\_count](#)
- int [auto\\_all\\_ht\\_revise\\_count\\_k](#)
- int [max\\_gen\\_rels](#)
- int [max\\_gen\\_scope](#)
- int [max\\_gen\\_tuples](#)
- int [max\\_cluster\\_tables](#)
- [llist](#) \* [usedhts\\_list](#)
- int [rs2gbtd](#)
- int [log](#)
- [llist](#) \* [all\\_tree\\_hts](#)
- int [node\\_merge](#)

### 3.31.1 Member Data Documentation

3.31.1.1 int [main\\_structure::ac3](#)

3.31.1.2 [llist](#)\* [main\\_structure::active\\_htables](#)

3.31.1.3 struct [alive\\_table](#)\* [main\\_structure::alive\\_t](#)

3.31.1.4 int [main\\_structure::all\\_solutions](#)

3.31.1.5 [llist](#)\* [main\\_structure::all\\_tree\\_hts](#)

3.31.1.6 int [main\\_structure::apply\\_td](#)

3.31.1.7 [alive\\_table](#)\* [main\\_structure::at](#)

3.31.1.8 int [main\\_structure::auto\\_all\\_ht\\_revise\\_count](#)

3.31.1.9 int [main\\_structure::auto\\_all\\_ht\\_revise\\_count\\_k](#)

3.31.1.10 int [main\\_structure::auto\\_all\\_revise\\_count](#)

3.31.1.11 int [main\\_structure::auto\\_one\\_ht\\_revise\\_count](#)

3.31.1.12 int [main\\_structure::auto\\_one\\_ht\\_revise\\_count\\_k](#)

3.31.1.13 int [main\\_structure::auto\\_one\\_revise\\_count](#)

3.31.1.14 int [main\\_structure::bayesian](#)

3.31.1.15 int [main\\_structure::bt\\_counter](#)

3.31.1.16 int [main\\_structure::bt\\_visit\\_counter](#)

3.31.1.17 struct [constraint\\_graph](#)\* [main\\_structure::cg](#)

3.31.1.18 int [main\\_structure::check\\_counter](#)

3.31.1.19 int\* [main\\_structure::cmap](#)

3.31.1.20 int [main\\_structure::comb\\_ids](#)

- 3.31.1.21 int main\_structure::comb\_size
- 3.31.1.22 int main\_structure::comb\_size\_search
- 3.31.1.23 constraints\* main\_structure::cons
- 3.31.1.24 int main\_structure::consistant\_print
- 3.31.1.25 int main\_structure::consistant\_solution\_count
- 3.31.1.26 int main\_structure::constraint\_placement
- 3.31.1.27 int main\_structure::construct\_solution
- 3.31.1.28 int main\_structure::db\_algo
- 3.31.1.29 int main\_structure::dcid
- 3.31.1.30 struct comb\_data\* main\_structure::delme
- 3.31.1.31 struct decomposition\_info\* main\_structure::di
- 3.31.1.32 llist\* main\_structure::domains
- 3.31.1.33 struct decision\_tree\_system\* main\_structure::dts
- 3.31.1.34 llist\* main\_structure::effected\_vars\_by\_gac
- 3.31.1.35 int main\_structure::expression\_pointer
- 3.31.1.36 int\* main\_structure::expressions
- 3.31.1.37 int main\_structure::fc\_cbj\_fc
- 3.31.1.38 int main\_structure::feature\_vector\_options
- 3.31.1.39 char\* main\_structure::filename
- 3.31.1.40 int main\_structure::forward\_check
- 3.31.1.41 int main\_structure::function\_offset
- 3.31.1.42 int main\_structure::h
- 3.31.1.43 int main\_structure::ht\_id
- 3.31.1.44 int main\_structure::init\_count
- 3.31.1.45 int main\_structure::just\_combinations
- 3.31.1.46 int main\_structure::just\_redundancy\_removal
- 3.31.1.47 int main\_structure::left\_deep
- 3.31.1.48 int main\_structure::log

- 3.31.1.49 int main\_structure::long\_tuple\_index
- 3.31.1.50 int main\_structure::m
- 3.31.1.51 int main\_structure::max\_cluster\_tables
- 3.31.1.52 int main\_structure::max\_domain\_value
- 3.31.1.53 int main\_structure::max\_gen\_rels
- 3.31.1.54 int main\_structure::max\_gen\_scope
- 3.31.1.55 int main\_structure::max\_gen\_tuples
- 3.31.1.56 int main\_structure::minibucket
- 3.31.1.57 int main\_structure::minibucketopts
- 3.31.1.58 struct main\_structure\* main\_structure::next
- 3.31.1.59 options main\_structure::no\_check\_insol\_tuples
- 3.31.1.60 int main\_structure::no\_support
- 3.31.1.61 int main\_structure::no\_tuples
- 3.31.1.62 int main\_structure::node\_merge
- 3.31.1.63 int main\_structure::nosingleedge
- 3.31.1.64 long double main\_structure::number\_of\_solutions
- 3.31.1.65 int main\_structure::only\_check\_connected
- 3.31.1.66 int main\_structure::partition\_id
- 3.31.1.67 int main\_structure::pprs
- 3.31.1.68 struct main\_structure\* main\_structure::previous
- 3.31.1.69 int main\_structure::print\_chk
- 3.31.1.70 char\* main\_structure::problemname
- 3.31.1.71 int main\_structure::profile
- 3.31.1.72 llist\* main\_structure::profile\_times
- 3.31.1.73 int main\_structure::project
- 3.31.1.74 options main\_structure::queue
- 3.31.1.75 int main\_structure::r
- 3.31.1.76 options main\_structure::r\_star\_algo

- 3.31.1.77 **options** main\_structure::r\_star\_all\_single\_sol
- 3.31.1.78 **options** main\_structure::r\_star\_gac
- 3.31.1.79 **int** main\_structure::rel\_revise\_counter
- 3.31.1.80 **relation\*** main\_structure::relations
- 3.31.1.81 **int** main\_structure::remaining\_nodes
- 3.31.1.82 **int** main\_structure::remove\_redundant\_edges
- 3.31.1.83 **int** main\_structure::rpwc
- 3.31.1.84 **int** main\_structure::rs2gbtd
- 3.31.1.85 **options** main\_structure::rs2tree\_scheme
- 3.31.1.86 **options** main\_structure::rs2tree\_scheme\_options
- 3.31.1.87 **int** main\_structure::rs2tree\_scheme\_options\_parameter
- 3.31.1.88 **options** main\_structure::search
- 3.31.1.89 **struct solution\*** main\_structure::sol
- 3.31.1.90 **int** main\_structure::solve
- 3.31.1.91 **int** main\_structure::split\_cutoff
- 3.31.1.92 **clock\_t** main\_structure::split\_time
- 3.31.1.93 **double** main\_structure::startt
- 3.31.1.94 **int\*\*** main\_structure::temp
- 3.31.1.95 **int** main\_structure::temp\_count
- 3.31.1.96 **int\*** main\_structure::temp\_int\_p
- 3.31.1.97 **void\*** main\_structure::temp\_p
- 3.31.1.98 **char\*** main\_structure::tempstr
- 3.31.1.99 **llist\*** main\_structure::tms\_list
- 3.31.1.100 **int** main\_structure::tms\_total\_inserted
- 3.31.1.101 **llist\*** main\_structure::to\_queue\_supports
- 3.31.1.102 **int** main\_structure::total\_nodes
- 3.31.1.103 **int** main\_structure::trace\_trigger
- 3.31.1.104 **int** main\_structure::tree\_bt\_time

- 3.31.1.105 options main\_structure::tree\_decom
- 3.31.1.106 options main\_structure::tree\_decom\_solve
- 3.31.1.107 options main\_structure::tree\_decom\_solve\_1\_all
- 3.31.1.108 int main\_structure::tree\_decom\_wb
- 3.31.1.109 set\* main\_structure::updated\_combinations
- 3.31.1.110 int main\_structure::use\_histo
- 3.31.1.111 llist\* main\_structure::usedhts\_list
- 3.31.1.112 int main\_structure::v
- 3.31.1.113 int main\_structure::valid
- 3.31.1.114 int main\_structure::variable\_choice\_huristic
- 3.31.1.115 variables\* main\_structure::vars
- 3.31.1.116 int main\_structure::visit\_counter
- 3.31.1.117 int\* main\_structure::vmap
- 3.31.1.118 llist\* main\_structure::vmap\_dirty\_list
- 3.31.1.119 int main\_structure::with\_index\_delete\_updating
- 3.31.1.120 int main\_structure::with\_tuple\_support\_recording

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

- [globals.h](#)

## 3.32 node\_graph Struct Reference

```
#include <decomposition.h>
```

### Public Attributes

- int vid
- llist \*\* adjacency\_list
- char \*\* matrix
- char out\_of\_order
- int peo
- int fill\_in
- int heap\_pos
- int marked\_neighbours
- int node\_count
- int \* node\_map

### 3.32.1 Member Data Documentation

3.32.1.1 `llist** node_graph::adjacency_list`

3.32.1.2 `int node_graph::fill_in`

3.32.1.3 `int node_graph::heap_pos`

3.32.1.4 `int node_graph::marked_neighbours`

3.32.1.5 `char** node_graph::matrix`

3.32.1.6 `int node_graph::node_count`

3.32.1.7 `int* node_graph::node_map`

3.32.1.8 `char node_graph::out_of_order`

3.32.1.9 `int node_graph::peo`

3.32.1.10 `int node_graph::vid`

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

- [decomposition.h](#)

## 3.33 parameter Struct Reference

```
#include <constraints.h>
```

### Public Attributes

- `int value_type`
- `void * value`

### 3.33.1 Member Data Documentation

3.33.1.1 `void* parameter::value`

3.33.1.2 `int parameter::value_type`

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

- [constraints.h](#)

## 3.34 pseudo\_cluster Struct Reference

```
#include <gbtd.h>
```

## Public Attributes

- int `id`
- int `cluster_id`
- `llist` \* `unassigned_vars`
- `llist` \* `assigned_vars`
- `llist` \* `undo_info`
- `llist` \* `children`
- int `unassigned_count`
- double `solCount`
- int `isolCount`
- struct `pseudo_cluster` \* `parent`
- `llist_node` \* `parent_list_self`
- int `dfs_state`
- `decomposed_tree` \* `treen`
- int `all_sol_state`
- `llist` \* `separator_vars`
- `tree_map` \* `goods`
- `tree_map` \* `nogoods`
- `llist` \* `sep_commons`

### 3.34.1 Member Data Documentation

3.34.1.1 int `pseudo_cluster::all_sol_state`

3.34.1.2 `llist`\* `pseudo_cluster::assigned_vars`

3.34.1.3 `llist`\* `pseudo_cluster::children`

3.34.1.4 int `pseudo_cluster::cluster_id`

3.34.1.5 int `pseudo_cluster::dfs_state`

3.34.1.6 `tree_map`\* `pseudo_cluster::goods`

3.34.1.7 int `pseudo_cluster::id`

3.34.1.8 int `pseudo_cluster::isolCount`

3.34.1.9 `tree_map`\* `pseudo_cluster::nogoods`

3.34.1.10 struct `pseudo_cluster`\* `pseudo_cluster::parent`

3.34.1.11 `llist_node`\* `pseudo_cluster::parent_list_self`

3.34.1.12 `llist`\* `pseudo_cluster::sep_commons`

3.34.1.13 `llist`\* `pseudo_cluster::separator_vars`

3.34.1.14 double `pseudo_cluster::solCount`

3.34.1.15 `decomposed_tree`\* `pseudo_cluster::treen`

3.34.1.16 int `pseudo_cluster::unassigned_count`

3.34.1.17 `llist* pseudo_cluster::unassigned_vars`

3.34.1.18 `llist* pseudo_cluster::undo_info`

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

- [gbtd.h](#)

## 3.35 `rb_tree` Struct Reference

```
#include <rb_tree.h>
```

### Public Attributes

- `int id`
- `char color`
- `void * key`
- `int ikey`
- `void * payload`
- `struct rb_tree * parent`
- `struct rb_tree * left`
- `struct rb_tree * right`

### 3.35.1 Member Data Documentation

3.35.1.1 `char rb_tree::color`

3.35.1.2 `int rb_tree::id`

3.35.1.3 `int rb_tree::ikey`

3.35.1.4 `void* rb_tree::key`

3.35.1.5 `struct rb_tree* rb_tree::left`

3.35.1.6 `struct rb_tree* rb_tree::parent`

3.35.1.7 `void* rb_tree::payload`

3.35.1.8 `struct rb_tree* rb_tree::right`

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

- [rb\\_tree.h](#)

## 3.36 `rbt_set` Struct Reference

```
#include <rbt_set.h>
```



## Public Attributes

- int [size](#)
- int [count](#)
- [rb\\_tree](#) \* [tmap](#)

### 3.36.1 Member Data Documentation

3.36.1.1 int [rbt\\_set::count](#)

3.36.1.2 int [rbt\\_set::size](#)

3.36.1.3 [rb\\_tree](#)\* [rbt\\_set::tmap](#)

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

- [rbt\\_set.h](#)

## 3.37 rs2tree\_configuration Struct Reference

```
#include <rs2tree.h>
```

## Public Attributes

- [decomposed\\_tree](#) \*\* [treens](#)
- [llist](#) \* [usedhts\\_list](#)
- [set](#) \* [queue](#)
- [set](#) \*\* [queues](#)
- int [queues\\_count](#)

### 3.37.1 Member Data Documentation

3.37.1.1 [set](#)\* [rs2tree\\_configuration::queue](#)

3.37.1.2 [set](#)\*\* [rs2tree\\_configuration::queues](#)

3.37.1.3 int [rs2tree\\_configuration::queues\\_count](#)

3.37.1.4 [decomposed\\_tree](#)\*\* [rs2tree\\_configuration::treens](#)

3.37.1.5 [llist](#)\* [rs2tree\\_configuration::usedhts\\_list](#)

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

- [rs2tree.h](#)

## 3.38 rs2tree\_stats Struct Reference

```
#include <rs2tree.h>
```

## Public Attributes

- int `id`
- `l`list \* `varlist`
- `l`list \* `constraints`
- `l`list \* `sorted_overlaps`
- `l`list \* `sorted_overlaps_keys`
- `l`list \* `all_relations`
- int `sum0`
- int `deleted0`
- double `domain_product0`
- double `log_domain_sum`
- double `log_tuples_sum`
- int `sum1`
- int `deleted1`
- double `time`
- `decomposed_tree` \* `tn`
- double `f_factor`
- double `sol_count`
- double `avg_var_per_rel`
- double `std_var_per_rel`
- double `avg_tuple_per_vvp`
- double `std_tuple_per_vvp`
- double `avg_tuple_per_vvp_p`
- double `std_tuple_per_vvp_p`
- double `avg_tuple_per_vvp_norm`
- double `std_tuple_per_vvp_norm`
- double `avg_tuple_per_vvp_norm_p`
- double `std_tuple_per_vvp_norm_p`
- double `avg_tuple_per_vvp_log`
- double `std_tuple_per_vvp_log`
- double `avg_tuple_per_vvp_log_p`
- double `std_tuple_per_vvp_log_p`
- double `entropy_prod_avg`
- double `entropy_prod_std`
- double `entropy_sum`
- double `kappa`
- double `tuple_vvp_avg`
- double `tuple_vvp_std`
- double `tuple_vvp_avg_log`
- double `tuple_vvp_std_log`
- double `tuple_vvp_rto`
- int `relations`
- int `processed`

### 3.38.1 Member Data Documentation

3.38.1.1 `l`list\* `rs2tree_stats::all_relations`

3.38.1.2 double `rs2tree_stats::avg_tuple_per_vvp`

3.38.1.3 double `rs2tree_stats::avg_tuple_per_vvp_log`

3.38.1.4 double `rs2tree_stats::avg_tuple_per_vvp_log_p`

- 3.38.1.5 double rs2tree\_stats::avg\_tuple\_per\_vvp\_norm
- 3.38.1.6 double rs2tree\_stats::avg\_tuple\_per\_vvp\_norm\_p
- 3.38.1.7 double rs2tree\_stats::avg\_tuple\_per\_vvp\_p
- 3.38.1.8 double rs2tree\_stats::avg\_var\_per\_rel
- 3.38.1.9 ~~l~~list\* rs2tree\_stats::constraints
- 3.38.1.10 int rs2tree\_stats::deleted0
- 3.38.1.11 int rs2tree\_stats::deleted1
- 3.38.1.12 double rs2tree\_stats::domain\_product0
- 3.38.1.13 double rs2tree\_stats::entropy\_prod\_avg
- 3.38.1.14 double rs2tree\_stats::entropy\_prod\_std
- 3.38.1.15 double rs2tree\_stats::entropy\_sum
- 3.38.1.16 double rs2tree\_stats::f\_factor
- 3.38.1.17 int rs2tree\_stats::id
- 3.38.1.18 double rs2tree\_stats::kappa
- 3.38.1.19 double rs2tree\_stats::log\_domain\_sum
- 3.38.1.20 double rs2tree\_stats::log\_tuples\_sum
- 3.38.1.21 int rs2tree\_stats::processed
- 3.38.1.22 int rs2tree\_stats::relations
- 3.38.1.23 double rs2tree\_stats::sol\_count
- 3.38.1.24 ~~l~~list\* rs2tree\_stats::sorted\_overlaps
- 3.38.1.25 ~~l~~list\* rs2tree\_stats::sorted\_overlaps\_keys
- 3.38.1.26 double rs2tree\_stats::std\_tuple\_per\_vvp
- 3.38.1.27 double rs2tree\_stats::std\_tuple\_per\_vvp\_log
- 3.38.1.28 double rs2tree\_stats::std\_tuple\_per\_vvp\_log\_p
- 3.38.1.29 double rs2tree\_stats::std\_tuple\_per\_vvp\_norm
- 3.38.1.30 double rs2tree\_stats::std\_tuple\_per\_vvp\_norm\_p
- 3.38.1.31 double rs2tree\_stats::std\_tuple\_per\_vvp\_p
- 3.38.1.32 double rs2tree\_stats::std\_var\_per\_rel

- 3.38.1.33 int rs2tree\_stats::sum0
- 3.38.1.34 int rs2tree\_stats::sum1
- 3.38.1.35 double rs2tree\_stats::time
- 3.38.1.36 decomposed\_tree\* rs2tree\_stats::tn
- 3.38.1.37 double rs2tree\_stats::tuple\_vvp\_avg
- 3.38.1.38 double rs2tree\_stats::tuple\_vvp\_avg\_log
- 3.38.1.39 double rs2tree\_stats::tuple\_vvp\_rto
- 3.38.1.40 double rs2tree\_stats::tuple\_vvp\_std
- 3.38.1.41 double rs2tree\_stats::tuple\_vvp\_std\_log
- 3.38.1.42 llist\* rs2tree\_stats::varlist

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

- [rs2tree.h](#)

### 3.39 s\_node Struct Reference

```
#include <set.h>
```

#### Public Attributes

- int [key](#)
- void \* [body](#)

#### 3.39.1 Member Data Documentation

3.39.1.1 void\* s\_node::body

3.39.1.2 int s\_node::key

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

- [set.h](#)

### 3.40 separator\_info Struct Reference

```
#include <decomposition.h>
```

#### Public Attributes

- int [rel\\_count](#)
- htable \*\* [relations](#)

- [htable\\*\\* relation\\_mapped\\_to](#)
- [rb\\_tree\\* tables](#)

### 3.40.1 Member Data Documentation

3.40.1.1 `int separator_info::rel_count`

3.40.1.2 `htable** separator_info::relation_mapped_to`

3.40.1.3 `htable** separator_info::relations`

3.40.1.4 `rb_tree* separator_info::tables`

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

- [decomposition.h](#)

## 3.41 set Struct Reference

```
#include <set.h>
```

### Public Attributes

- [llist\\* list](#)
- [char\\* map](#)
- [llist\\_node\\*\\* element\\_ptrs](#)
- [int last\\_removed](#)
- [int size](#)
- [struct rb\\_tree\\* tmap](#)

### 3.41.1 Member Data Documentation

3.41.1.1 `llist_node** set::element_ptrs`

3.41.1.2 `int set::last_removed`

3.41.1.3 `llist* set::list`

3.41.1.4 `char* set::map`

3.41.1.5 `int set::size`

3.41.1.6 `struct rb_tree* set::tmap`

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

- [set.h](#)

## 3.42 solution Struct Reference

```
#include <solution.h>
```

## Public Attributes

- [int \\* values](#)

### 3.42.1 Member Data Documentation

#### 3.42.1.1 [int\\* solution::values](#)

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

- [solution.h](#)

## 3.43 [solution\\_structure](#) Struct Reference

```
#include <combinations.h>
```

## Public Attributes

- [int size](#)
- [llist \\*\\* solutions](#)

### 3.43.1 Member Data Documentation

#### 3.43.1.1 [int solution\\_structure::size](#)

#### 3.43.1.2 [llist\\*\\* solution\\_structure::solutions](#)

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

- [combinations.h](#)

## 3.44 [table\\_map](#) Struct Reference

```
#include <rs2.h>
```

## Public Attributes

- [int age](#)
- [char \\*\\* map](#)

### 3.44.1 Member Data Documentation

#### 3.44.1.1 [int table\\_map::age](#)

#### 3.44.1.2 [char\\*\\* table\\_map::map](#)

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

- [rs2.h](#)

## 3.45 tree\_map Struct Reference

```
#include <tree_map.h>
```

### Public Attributes

- [l](#)list \* [i](#)ndexed\_cols
- [h](#)table \* [h](#)tt
- [l](#)list \* [l](#)ists
- struct [tree\\_map\\_node](#) \* [a](#)lists
- int [i](#)d
- int [c](#)ount
- int [c](#)count
- char [d](#)eleted
- char [c](#)ompressed

### 3.45.1 Member Data Documentation

3.45.1.1 [struct tree\\_map\\_node\\*](#) [tree\\_map::a](#)lists

3.45.1.2 [int](#) [tree\\_map::c](#)count

3.45.1.3 [char](#) [tree\\_map::c](#)ompressed

3.45.1.4 [int](#) [tree\\_map::c](#)ount

3.45.1.5 [char](#) [tree\\_map::d](#)eleted

3.45.1.6 [htable\\*](#) [tree\\_map::h](#)tt

3.45.1.7 [int](#) [tree\\_map::i](#)d

3.45.1.8 [l](#)list\* [tree\\_map::i](#)ndexed\_cols

3.45.1.9 [l](#)list\* [tree\\_map::l](#)ists

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

- [tree\\_map.h](#)

## 3.46 tree\_map\_node Struct Reference

```
#include <tree_map.h>
```

### Public Attributes

- int [i](#)d
- union {
  - [l](#)list \* [i](#)nnexts
  - struct [tree\\_map\\_node](#) \* [a](#)nnexts
  - int \*\* [a](#)tuples
- [n](#)exts

- char [full](#)
- int [ccount](#)

### 3.46.1 Member Data Documentation

3.46.1.1 struct `tree_map_node*` `tree_map_node::anexts`

3.46.1.2 int\*\* `tree_map_node::atuples`

3.46.1.3 int `tree_map_node::ccount`

3.46.1.4 char `tree_map_node::full`

3.46.1.5 int `tree_map_node::id`

3.46.1.6 llist\* `tree_map_node::lnexts`

3.46.1.7 union { ... } `tree_map_node::nexts`

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

- [tree\\_map.h](#)

## 3.47 tree\_maps Struct Reference

```
#include <active_tuple.h>
```

### Public Attributes

- int [trees](#)
- `tree_map` \*\* [tms](#)
- int \* [histogram](#)
- `circular_array` \* [ca](#)
- int [free](#)
- int \* [inserted](#)
- int [max](#)

### 3.47.1 Member Data Documentation

3.47.1.1 `circular_array*` `tree_maps::ca`

3.47.1.2 int `tree_maps::free`

3.47.1.3 int\* `tree_maps::histogram`

3.47.1.4 int\* `tree_maps::inserted`

3.47.1.5 int `tree_maps::max`

3.47.1.6 `tree_map`\*\* `tree_maps::tms`



### 3.47.1.7 int tree\_maps::trees

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

- [active\\_tuple.h](#)

## 3.48 triplet Struct Reference

```
#include <rpwc.h>
```

### Public Attributes

- int \* [ttt](#)

### 3.48.1 Member Data Documentation

#### 3.48.1.1 int\* triplet::ttt

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

- [rpwc.h](#)

## 3.49 tuple\_ids Struct Reference

```
#include <hashtable.h>
```

### Public Attributes

- int \* [tuple](#)
- int \* [ids](#)
- int [deletet](#)

### 3.49.1 Member Data Documentation

#### 3.49.1.1 int tuple\_ids::deletet

#### 3.49.1.2 int\* tuple\_ids::ids

#### 3.49.1.3 int\* tuple\_ids::tuple

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

- [hashtable.h](#)

## 3.50 undo\_info Struct Reference

```
#include <rpwc.h>
```

## Public Attributes

- [llist \\*\\* reductions](#)
- [set \\* reduction\\_set](#)
- [undo\\_lastGAC \\* undo\\_LG](#)

### 3.50.1 Member Data Documentation

3.50.1.1 [set \\* undo\\_info::reduction\\_set](#)

3.50.1.2 [llist \\*\\* undo\\_info::reductions](#)

3.50.1.3 [undo\\_lastGAC \\* undo\\_info::undo\\_LG](#)

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

- [rpwc.h](#)

## 3.51 [undo\\_lastGAC](#) Struct Reference

```
#include <rpwc.h>
```

### Public Attributes

- [int \\*\\*\\* vars\\_cons\\_values](#)
- [llist \\* to\\_undo](#)

### 3.51.1 Member Data Documentation

3.51.1.1 [llist \\* undo\\_lastGAC::to\\_undo](#)

3.51.1.2 [int \\*\\*\\* undo\\_lastGAC::vars\\_cons\\_values](#)

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

- [rpwc.h](#)

## 3.52 [value](#) Struct Reference

```
#include <domains.h>
```

### Public Attributes

- [int value](#)
- [int position](#)
- [int killed](#)
- [int sacremoved](#)
- [char \\* name](#)
- [llist\\_node \\* domain\\_list\\_location](#)

### 3.52.1 Member Data Documentation

3.52.1.1 `llist_node*` `value::domain_list_location`

3.52.1.2 `int` `value::killed`

3.52.1.3 `char*` `value::name`

3.52.1.4 `int` `value::position`

3.52.1.5 `int` `value::sacremoved`

3.52.1.6 `int` `value::value`

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

- [domains.h](#)

## 3.53 var\_undo\_data Struct Reference

```
#include <gbtd.h>
```

### Public Attributes

- `set *` `relations`
- `light_stack *` `tuples`
- `light_stack *` `domains`
- `variable *` `var`

### 3.53.1 Member Data Documentation

3.53.1.1 `light_stack*` `var_undo_data::domains`

3.53.1.2 `set*` `var_undo_data::relations`

3.53.1.3 `light_stack*` `var_undo_data::tuples`

3.53.1.4 `variable*` `var_undo_data::var`

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

- [gbtd.h](#)

## 3.54 var\_val Struct Reference

```
#include <variables.h>
```

### Public Attributes

- `variable *` `var`
- `llist_node *` `value`

### 3.54.1 Member Data Documentation

3.54.1.1 `llist_node* var_val::value`

3.54.1.2 `variable* var_val::var`

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

- [variables.h](#)

## 3.55 var\_val\_ac Struct Reference

### Public Attributes

- `variable * var`
- `llist_node * valn`

### 3.55.1 Member Data Documentation

3.55.1.1 `llist_node* var_val_ac::valn`

3.55.1.2 `variable* var_val_ac::var`

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

- [ac.c](#)

## 3.56 variable Struct Reference

```
#include <variables.h>
```

### Public Attributes

- `int variable_number`
- `int domain_size`
- `value ** domain`
- `llist * current_domain`
- `llist * current_domain_hidden`
- `llist_node * current_domain_hashmap`
- `set * current_domain_set`
- `llist * reductions`
- `llist_node * choosen_value`
- `llist * constraints`
- `int orig_constraint_count`
- `struct rb_tree * constraints_rb_tree`
- `llist * htables`
- `set * neighbours`
- `llist * future`
- `llist * past`
- `llist * removed`
- `llist * conflicts`

- int `init_order`
- const char \* `name`
- int `occurance_count`
- int `static_order_position`
- int `max_domain_value`
- `llist` \* `conflicts_to_remove`
- int `last_domain_update_time`
- `llist` \* `htreductions`
- int `time`
- int `assigned_count`
- struct `lastGAC` \*\*\* `last_gac`
- struct `undo_info` \* `ui`

### 3.56.1 Member Data Documentation

3.56.1.1 int `variable::assigned_count`

3.56.1.2 `llist_node`\* `variable::chosen_value`

3.56.1.3 `llist`\* `variable::conflicts`

3.56.1.4 `llist`\* `variable::conflicts_to_remove`

3.56.1.5 `llist`\* `variable::constraints`

3.56.1.6 struct `rb_tree`\* `variable::constraints_rb_tree`

3.56.1.7 `llist`\* `variable::current_domain`

3.56.1.8 `llist_node`\* `variable::current_domain_hashmap`

3.56.1.9 `llist`\* `variable::current_domain_hidden`

3.56.1.10 `set`\* `variable::current_domain_set`

3.56.1.11 `value`\*\* `variable::domain`

3.56.1.12 int `variable::domain_size`

3.56.1.13 `llist`\* `variable::future`

3.56.1.14 `llist`\* `variable::htables`

3.56.1.15 `llist`\* `variable::htreductions`

3.56.1.16 int `variable::init_order`

3.56.1.17 int `variable::last_domain_update_time`

3.56.1.18 struct `lastGAC`\*\*\* `variable::last_gac`

3.56.1.19 int `variable::max_domain_value`

3.56.1.20 const char\* `variable::name`

- 3.56.1.21 `set* variable::neighbours`
- 3.56.1.22 `int variable::occurance_count`
- 3.56.1.23 `int variable::orig_constraint_count`
- 3.56.1.24 `llist* variable::past`
- 3.56.1.25 `llist* variable::reductions`
- 3.56.1.26 `llist* variable::removed`
- 3.56.1.27 `int variable::static_order_position`
- 3.56.1.28 `int variable::time`
- 3.56.1.29 `struct undo_info* variable::ui`
- 3.56.1.30 `int variable::variable_number`

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

- [variables.h](#)

## 3.57 variables Struct Reference

```
#include <variables.h>
```

### Public Attributes

- `int count`
- `int size`
- `int assigned_count`
- `llist * variable_list`
- `llist * variable_list_init`
- `variable ** map_to_id`
- `int min_domain`
- `int max_domain`

### 3.57.1 Member Data Documentation

- 3.57.1.1 `int variables::assigned_count`
- 3.57.1.2 `int variables::count`
- 3.57.1.3 `variable** variables::map_to_id`
- 3.57.1.4 `int variables::max_domain`
- 3.57.1.5 `int variables::min_domain`
- 3.57.1.6 `int variables::size`

3.57.1.7 `llist*` `variables::variable_list`

3.57.1.8 `llist*` `variables::variable_list_init`

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

- [variables.h](#)





# Chapter 4

## File Documentation

### 4.1 ac.c File Reference

```
#include "ac.h"  
#include "set.h"  
#include "l1ist.h"  
#include "utils.h"
```

#### Classes

- struct [var\\_val\\_ac](#)

#### Typedefs

- typedef struct [var\\_val\\_ac](#) [var\\_val\\_ac](#)

#### Functions

- void [ac4](#) ([main\\_structure](#) \**m\_s*, int *print*, int *check*)
- int [ac3](#) (int *version*, [main\\_structure](#) \**m\_s*, int *print*, int *check*)
- void [check\\_arc\\_consist](#) ([main\\_structure](#) \**m\_s*)

#### 4.1.1 Typedef Documentation

4.1.1.1 typedef struct [var\\_val\\_ac](#) [var\\_val\\_ac](#)

#### 4.1.2 Function Documentation

4.1.2.1 int [ac3](#) ( int *version*, [main\\_structure](#) \* *m\_s*, int *print*, int *check* )

4.1.2.2 void [ac4](#) ( [main\\_structure](#) \* *m\_s*, int *print*, int *check* )

4.1.2.3 void [check\\_arc\\_consist](#) ( [main\\_structure](#) \* *m\_s* )

## 4.2 ac.h File Reference

```
#include <stdio.h>
#include "globals.h"
#include "variables.h"
#include "revise.h"
```

### Functions

- int [ac3](#) (int version, [main\\_structure](#) \*m\_s, int print, int check)
- void [ac4](#) ([main\\_structure](#) \*m\_s, int print, int check)
- void [check\\_arc\\_consist](#) ([main\\_structure](#) \*m\_s)

### 4.2.1 Detailed Description

Provides the implementation of arc consistency algorithms.

### 4.2.2 Function Documentation

4.2.2.1 int [ac3](#) ( int *version*, [main\\_structure](#) \* *m\_s*, int *print*, int *check* )

4.2.2.2 void [ac4](#) ( [main\\_structure](#) \* *m\_s*, int *print*, int *check* )

4.2.2.3 void [check\\_arc\\_consist](#) ( [main\\_structure](#) \* *m\_s* )

## 4.3 active\_tuple.c File Reference

```
#include "utils.h"
#include "active_tuple.h"
#include "decomposition.h"
#include "hashtable_search.h"
#include "rs2tree.h"
#include "circular_array.h"
#include <math.h>
```

### Functions

- [tree\\_maps](#) \* [new\\_tree\\_maps](#) (int trees, [llist](#) \*common\_cols)
- void [destroy\\_tree\\_maps](#) ([tree\\_maps](#) \*tms)
- [active\\_tuple](#) \* [new\\_active\\_tuple](#) (int number\_of\_vars, int length, [llist](#) \*lvid, [rb\\_tree](#) \*all\_ac\_vars)
- [active\\_tuple](#) \* [new\\_active\\_tuple\\_sizeable](#) (int number\_of\_vars, int length, [llist](#) \*lvid, int sizeable, [rb\\_tree](#) \*all\_ac\_vars)
- void [add\\_active\\_tuple](#) ([active\\_tuple](#) \*ac, int number\_of\_vars, int length, [llist](#) \*lvid)
- void [destroy\\_active\\_tuple](#) ([active\\_tuple](#) \*ac)
- void [extend\\_tuple](#) ([active\\_tuple](#) \*ac, [htable](#) \*ht, int t)
- void [extend\\_tuple\\_with\\_single\\_value](#) ([active\\_tuple](#) \*ac, [variable](#) \*var, int value)
- void [pop\\_tuple](#) ([active\\_tuple](#) \*ac)
- void [store\\_tuple](#) ([active\\_tuple](#) \*ac, int wildcard)
- void [store\\_tuple\\_payload](#) ([active\\_tuple](#) \*ac, int wildcard, void \*payload)
- void [reset\\_ac\\_tm](#) ([active\\_tuple](#) \*ac)

- int `get_stored_weight_for_tuple` (`active_tuple *ac`, `htable *ht`, int `t`, long double `*weight`)
- int `get_stored_weight_for_tuple_from_tree` (`active_tuple *ac`, `htable *ht`, int `t`, long double `*weight`, `tree_map *tm`)
- void `store_weight_for_tuple` (`active_tuple *ac`, `htable *ht`, int `t`, long double `weight`)

### 4.3.1 Function Documentation

4.3.1.1 void `add_active_tuple` ( `active_tuple * ac`, int `number_of_vars`, int `length`, `llist * lvid` )

4.3.1.2 void `destroy_active_tuple` ( `active_tuple * ac` )

4.3.1.3 void `destroy_tree_maps` ( `tree_maps * tms` )

4.3.1.4 void `extend_tuple` ( `active_tuple * ac`, `htable * ht`, int `t` )

4.3.1.5 void `extend_tuple_with_single_value` ( `active_tuple * ac`, `variable * var`, int `value` )

4.3.1.6 int `get_stored_weight_for_tuple` ( `active_tuple * ac`, `htable * ht`, int `t`, long double `* weight` )

4.3.1.7 int `get_stored_weight_for_tuple_from_tree` ( `active_tuple * ac`, `htable * ht`, int `t`, long double `* weight`, `tree_map * tm` )

4.3.1.8 `active_tuple*` `new_active_tuple` ( int `number_of_vars`, int `length`, `llist * lvid`, `rb_tree * all_ac_vars` )

4.3.1.9 `active_tuple*` `new_active_tuple_sizeable` ( int `number_of_vars`, int `length`, `llist * lvid`, int `sizeable`, `rb_tree * all_ac_vars` )

4.3.1.10 `tree_maps*` `new_tree_maps` ( int `trees`, `llist * common_cols` )

4.3.1.11 void `pop_tuple` ( `active_tuple * ac` )

4.3.1.12 void `reset_ac_tm` ( `active_tuple * ac` )

4.3.1.13 void `store_tuple` ( `active_tuple * ac`, int `wildcard` )

generating the table

end of creating the table/index

generating the table

end of creating the table/index

4.3.1.14 void `store_tuple_payload` ( `active_tuple * ac`, int `wildcard`, void `* payload` )

4.3.1.15 void `store_weight_for_tuple` ( `active_tuple * ac`, `htable * ht`, int `t`, long double `weight` )

## 4.4 active\_tuple.h File Reference

```
#include "tree_map.h"
#include "circular_array.h"
#include "rb_tree.h"
```

## Classes

- struct [tree\\_maps](#)
- struct [active\\_tuple](#)

## Typedefs

- typedef struct [tree\\_maps](#) [tree\\_maps](#)
- typedef struct [active\\_tuple](#) [active\\_tuple](#)

## Functions

- [tree\\_maps](#) \* [new\\_tree\\_maps](#) (int *trees*, [llist](#) \**common\_cols*)
- [active\\_tuple](#) \* [new\\_active\\_tuple\\_sizeable](#) (int *number\_of\_vars*, int *length*, [llist](#) \**lvid*, int *sizeable*, [rb\\_tree](#) \**all\_ac\_vars*)
- [active\\_tuple](#) \* [new\\_active\\_tuple](#) (int *number\_of\_vars*, int *length*, [llist](#) \**lvid*, [rb\\_tree](#) \**all\_ac\_vars*)
- void [destroy\\_active\\_tuple](#) ([active\\_tuple](#) \**ac*)
- void [destroy\\_tree\\_maps](#) ([tree\\_maps](#) \**tms*)
- void [extend\\_tuple](#) ([active\\_tuple](#) \**ac*, [htable](#) \**ht*, int *t*)
- void [pop\\_tuple](#) ([active\\_tuple](#) \**ac*)
- void [store\\_tuple\\_payload](#) ([active\\_tuple](#) \**ac*, int *wildcard*, void \**payload*)
- void [store\\_tuple](#) ([active\\_tuple](#) \**ac*, int *wildcard*)
- void [extend\\_tuple\\_with\\_single\\_value](#) ([active\\_tuple](#) \**ac*, [variable](#) \**var*, int *value*)
- void [add\\_active\\_tuple](#) ([active\\_tuple](#) \**ac*, int *number\_of\_vars*, int *length*, [llist](#) \**lvid*)
- void \* [set\\_tuple\\_int\\_weight](#) ([active\\_tuple](#) \**ac*, int *weight*)
- int [get\\_stored\\_weight\\_for\\_tuple\\_from\\_tree](#) ([active\\_tuple](#) \**ac*, [htable](#) \**ht*, int *t*, long double \**weight*, [tree\\_map](#) \**tm*)
- int [get\\_stored\\_weight\\_for\\_tuple](#) ([active\\_tuple](#) \**ac*, [htable](#) \**ht*, int *t*, long double \**weight*)
- void [store\\_weight\\_for\\_tuple](#) ([active\\_tuple](#) \**ac*, [htable](#) \**ht*, int *t*, long double *weight*)
- void [reset\\_ac\\_tm](#) ([active\\_tuple](#) \**ac*)

### 4.4.1 Detailed Description

Provides a tree structure representation of tuples and methods for constructing and querying them.

### 4.4.2 Typedef Documentation

#### 4.4.2.1 typedef struct [active\\_tuple](#) [active\\_tuple](#)

#### 4.4.2.2 typedef struct [tree\\_maps](#) [tree\\_maps](#)

### 4.4.3 Function Documentation

#### 4.4.3.1 void [add\\_active\\_tuple](#) ( [active\\_tuple](#) \* *ac*, int *number\_of\_vars*, int *length*, [llist](#) \* *lvid* )

#### 4.4.3.2 void [destroy\\_active\\_tuple](#) ( [active\\_tuple](#) \* *ac* )

#### 4.4.3.3 void [destroy\\_tree\\_maps](#) ( [tree\\_maps](#) \* *tms* )

#### 4.4.3.4 void [extend\\_tuple](#) ( [active\\_tuple](#) \* *ac*, [htable](#) \* *ht*, int *t* )

#### 4.4.3.5 void [extend\\_tuple\\_with\\_single\\_value](#) ( [active\\_tuple](#) \* *ac*, [variable](#) \* *var*, int *value* )

- 4.4.3.6 `int get_stored_weight_for_tuple ( active_tuple * ac, htable * ht, int t, long double * weight )`
- 4.4.3.7 `int get_stored_weight_for_tuple_from_tree ( active_tuple * ac, htable * ht, int t, long double * weight, tree_map * tm )`
- 4.4.3.8 `active_tuple* new_active_tuple ( int number_of_vars, int length, llist * lvid, rb_tree * all_ac_vars )`
- 4.4.3.9 `active_tuple* new_active_tuple_sizeable ( int number_of_vars, int length, llist * lvid, int sizeable, rb_tree * all_ac_vars )`
- 4.4.3.10 `tree_maps* new_tree_maps ( int trees, llist * common_cols )`
- 4.4.3.11 `void pop_tuple ( active_tuple * ac )`
- 4.4.3.12 `void reset_ac_tm ( active_tuple * ac )`
- 4.4.3.13 `void* set_tuple_int_weight ( active_tuple * ac, int weight )`
- 4.4.3.14 `void store_tuple ( active_tuple * ac, int wildcard )`

generating the table

end of creating the table/index

generating the table

end of creating the table/index

- 4.4.3.15 `void store_tuple_payload ( active_tuple * ac, int wildcard, void * payload )`
- 4.4.3.16 `void store_weight_for_tuple ( active_tuple * ac, htable * ht, int t, long double weight )`

## 4.5 btdallsol.c File Reference

```
#include "btdallsol.h"
#include "rpscprocedures.h"
#include "rs2tree.h"
#include "rs2.h"
```

### Functions

- `int find_match_good_nogood (decomposed_tree *child, int goods, long double *weight)`
- `int find_match_good_nogood_from_tree (decomposed_tree *child, int goods, long double *weight, tree_map *tm)`
- `int update_good_weight_from_tree (decomposed_tree *child, tree_map *tm, long double weight)`
- `void update_good_weight (decomposed_tree *child, long double weight)`
- `int btd (btd_engine btde, decomposed_tree *root, int recid, long double *consistent_sols)`
- `void init_btd_engine (btd_engine btde, decomposed_tree *root)`
- `void init_btd_tree (decomposed_tree *treen)`
- `btd_data * new_btd_data (decomposed_tree *treen)`
- `btd_data * new_btd_data2 (decomposed_tree *treen, int sizeable)`
- `variable * btd_label (variable *var_i, int *consistent, long double *consistent_count, btd_engine btde, decomposed_tree *root, int recid)`
- `variable * btd_unlabel (variable *var_i, int *consistent, main_structure *m_s)`
- `variable * choose_variable_from_tree (btd_engine btde)`

- void `print_separator` (`decomposed_tree` \**treen*, int *goods*)
- void `print_separator_h` (`decomposed_tree` \**treen*, int *goods*)
- int `btd2` (`decomposed_tree` \**treen*, long double \**total\_sols\_ret*)
- int `btd2_rec` (`decomposed_tree` \**treen*, long double \**total\_sols\_ret*)
- `variable` \* `btd_un_assign_variable` (`light_stack` \**assigned\_var\_stack*, `light_stack` \**undo\_value\_stack*)
- void `btd_undo_value_deletions` (`light_stack` \**value\_stack*)
- int `btd_assign_variable` (`variable` \**var*)
- void `btd_prepare_tree_for_btd2` (`decomposed_tree` \**treen*)
- void `btd_reset_variable_domain` (`variable` \**var*)
- void `btd_reset_variable_domain2` (`variable` \**var*)

#### 4.5.1 Function Documentation

- 4.5.1.1 int `btd` ( `btd_engine` *btde*, `decomposed_tree` \* *root*, int *recid*, long double \* *consistent\_sols* )
- 4.5.1.2 int `btd2` ( `decomposed_tree` \* *treen*, long double \* *total\_sols\_ret* )
- 4.5.1.3 int `btd2_rec` ( `decomposed_tree` \* *treen*, long double \* *total\_sols\_ret* )
- 4.5.1.4 int `btd_assign_variable` ( `variable` \* *var* )
- 4.5.1.5 `variable`\* `btd_label` ( `variable` \* *var\_i*, int \* *consistent*, long double \* *consistent\_count*, `btd_engine` *btde*, `decomposed_tree` \* *root*, int *recid* )
- 4.5.1.6 void `btd_prepare_tree_for_btd2` ( `decomposed_tree` \* *treen* )
- 4.5.1.7 void `btd_reset_variable_domain` ( `variable` \* *var* )
- 4.5.1.8 void `btd_reset_variable_domain2` ( `variable` \* *var* )
- 4.5.1.9 `variable`\* `btd_un_assign_variable` ( `light_stack` \* *assigned\_var\_stack*, `light_stack` \* *undo\_value\_stack* )
- 4.5.1.10 void `btd_undo_value_deletions` ( `light_stack` \* *value\_stack* )
- 4.5.1.11 `variable`\* `btd_unlabel` ( `variable` \* *var\_i*, int \* *consistent*, `main_structure` \* *m\_s* )
- 4.5.1.12 `variable`\* `choose_variable_from_tree` ( `btd_engine` *btde* )
- 4.5.1.13 int `find_match_good_nogood` ( `decomposed_tree` \* *child*, int *goods*, long double \* *weight* )
- 4.5.1.14 int `find_match_good_nogood_from_tree` ( `decomposed_tree` \* *child*, int *goods*, long double \* *weight*, `tree_map` \* *tm* )
- 4.5.1.15 void `init_btd_engine` ( `btd_engine` *btde*, `decomposed_tree` \* *root* )
- 4.5.1.16 void `init_btd_tree` ( `decomposed_tree` \* *treen* )
- 4.5.1.17 `btd_data`\* `new_btd_data` ( `decomposed_tree` \* *treen* )
- 4.5.1.18 `btd_data`\* `new_btd_data2` ( `decomposed_tree` \* *treen*, int *sizeable* )
- 4.5.1.19 void `print_separator` ( `decomposed_tree` \* *treen*, int *goods* )
- 4.5.1.20 void `print_separator_h` ( `decomposed_tree` \* *treen*, int *goods* )

4.5.1.21 void update\_good\_weight ( decomposed\_tree \* child, long double weight )

4.5.1.22 int update\_good\_weight\_from\_tree ( decomposed\_tree \* child, tree\_map \* tm, long double weight )

## 4.6 btdallsol.h File Reference

```
#include "utils.h"
#include "l1list.h"
#include "active_tuple.h"
#include "decomposition.h"
```

### Classes

- struct [btd\\_data](#)
- struct [btd\\_engine](#)

### Typedefs

- typedef struct [btd\\_data](#) [btd\\_data](#)
- typedef struct [btd\\_engine](#) [btd\\_engine](#)

### Functions

- int [find\\_match\\_good\\_nogood\\_from\\_tree](#) (decomposed\_tree \*child, int goods, long double \*weight, tree\_map \*tm)
- int [find\\_match\\_good\\_nogood](#) (decomposed\_tree \*child, int goods, long double \*weight)
- int [btd](#) (btd\_engine btde, decomposed\_tree \*root, int recid, long double \*consistent\_sols)
- void [init\\_btd\\_engine](#) (btd\_engine btde, decomposed\_tree \*root)
- void [init\\_btd\\_tree](#) (decomposed\_tree \*treen)
- [btd\\_data](#) \* [new\\_btd\\_data2](#) (decomposed\_tree \*treen, int sizeable)
- [btd\\_data](#) \* [new\\_btd\\_data](#) (decomposed\_tree \*treen)
- int [btd\\_bcssp](#) (decomposed\_tree \*treen, btd\_engine btde)
- [variable](#) \* [btd\\_label](#) ([variable](#) \*var\_i, int \*consistent, long double \*consistent\_count, [btd\\_engine](#) btde, decomposed\_tree \*root, int recid)
- [variable](#) \* [btd\\_unlabel](#) ([variable](#) \*var\_i, int \*consistent, [main\\_structure](#) \*m\_s)
- [variable](#) \* [choose\\_variable\\_from\\_tree](#) (btd\_engine btde)
- void [print\\_separator](#) (decomposed\_tree \*treen, int goods)
- void [print\\_separator\\_h](#) (decomposed\_tree \*treen, int goods)
- int [btd2\\_rec](#) (decomposed\_tree \*treen, long double \*total\_sols\_ret)
- int [btd2](#) (decomposed\_tree \*treen, long double \*total\_sols\_ret)
- [variable](#) \* [btd\\_un\\_assign\\_variable](#) ([light\\_stack](#) \*assigned\_var\_stack, [light\\_stack](#) \*undo\_value\_stack)
- void [btd\\_undo\\_value\\_deletions](#) ([light\\_stack](#) \*undo\_value\_stack)
- int [btd\\_assign\\_variable](#) ([variable](#) \*var)
- void [btd\\_prepare\\_tree\\_for\\_btd2](#) (decomposed\_tree \*treen)
- void [btd\\_reset\\_variable\\_domain](#) ([variable](#) \*var)
- void [btd\\_reset\\_variable\\_domain2](#) ([variable](#) \*var)
- void [update\\_good\\_weight](#) (decomposed\_tree \*child, long double weight)
- int [update\\_good\\_weight\\_from\\_tree](#) (decomposed\_tree \*child, tree\_map \*tm, long double weight)

#### 4.6.1 Detailed Description

Provides the implementation for finding all solutions using BTd.

## 4.6.2 Typedef Documentation

4.6.2.1 typedef struct *btd\_data* *btd\_data*

4.6.2.2 typedef struct *btd\_engine* *btd\_engine*

## 4.6.3 Function Documentation

4.6.3.1 int *btd* ( *btd\_engine* *btde*, *decomposed\_tree* \* *root*, int *recid*, long double \* *consistent\_sols* )

4.6.3.2 int *btd2* ( *decomposed\_tree* \* *treen*, long double \* *total\_sols\_ret* )

4.6.3.3 int *btd2\_rec* ( *decomposed\_tree* \* *treen*, long double \* *total\_sols\_ret* )

4.6.3.4 int *btd\_assign\_variable* ( *variable* \* *var* )

4.6.3.5 int *btd\_bcssp* ( *decomposed\_tree* \* *treen*, *btd\_engine* *btde* )

4.6.3.6 *variable*\* *btd\_label* ( *variable* \* *var\_i*, int \* *consistent*, long double \* *consistent\_count*, *btd\_engine* *btde*, *decomposed\_tree* \* *root*, int *recid* )

4.6.3.7 void *btd\_prepare\_tree\_for\_btd2* ( *decomposed\_tree* \* *treen* )

4.6.3.8 void *btd\_reset\_variable\_domain* ( *variable* \* *var* )

4.6.3.9 void *btd\_reset\_variable\_domain2* ( *variable* \* *var* )

4.6.3.10 *variable*\* *btd\_un\_assign\_variable* ( *light\_stack* \* *assigned\_var\_stack*, *light\_stack* \* *undo\_value\_stack* )

4.6.3.11 void *btd\_undo\_value\_deletions* ( *light\_stack* \* *undo\_value\_stack* )

4.6.3.12 *variable*\* *btd\_unlabel* ( *variable* \* *var\_i*, int \* *consistent*, *main\_structure* \* *m\_s* )

4.6.3.13 *variable*\* *choose\_variable\_from\_tree* ( *btd\_engine* *btde* )

4.6.3.14 int *find\_match\_good\_nogood* ( *decomposed\_tree* \* *child*, int *goods*, long double \* *weight* )

4.6.3.15 int *find\_match\_good\_nogood\_from\_tree* ( *decomposed\_tree* \* *child*, int *goods*, long double \* *weight*, *tree\_map* \* *tm* )

4.6.3.16 void *init\_btd\_engine* ( *btd\_engine* *btde*, *decomposed\_tree* \* *root* )

4.6.3.17 void *init\_btd\_tree* ( *decomposed\_tree* \* *treen* )

4.6.3.18 *btd\_data*\* *new\_btd\_data* ( *decomposed\_tree* \* *treen* )

4.6.3.19 *btd\_data*\* *new\_btd\_data2* ( *decomposed\_tree* \* *treen*, int *sizeable* )

4.6.3.20 void *print\_separator* ( *decomposed\_tree* \* *treen*, int *goods* )

4.6.3.21 void *print\_separator\_h* ( *decomposed\_tree* \* *treen*, int *goods* )

4.6.3.22 void *update\_good\_weight* ( *decomposed\_tree* \* *child*, long double *weight* )

4.6.3.23 int *update\_good\_weight\_from\_tree* ( *decomposed\_tree* \* *child*, *tree\_map* \* *tm*, long double *weight* )



## 4.7 circular\_array.c File Reference

```
#include "utils.h"  
#include "circular_array.h"
```

### Functions

- [circular\\_array \\* new\\_circular\\_array](#) (int size)
- int [add\\_to\\_circular\\_array](#) (int item, [circular\\_array \\*ca](#))
- void [destroy\\_circular\\_array](#) ([circular\\_array \\*ca](#))

#### 4.7.1 Function Documentation

4.7.1.1 int [add\\_to\\_circular\\_array](#) ( int *item*, [circular\\_array \\* ca](#) )

4.7.1.2 void [destroy\\_circular\\_array](#) ( [circular\\_array \\* ca](#) )

4.7.1.3 [circular\\_array\\*](#) [new\\_circular\\_array](#) ( int *size* )

## 4.8 circular\_array.h File Reference

### Classes

- struct [circular\\_array](#)

### Typedefs

- typedef struct [circular\\_array](#) [circular\\_array](#)

### Functions

- [circular\\_array \\* new\\_circular\\_array](#) (int size)
- int [add\\_to\\_circular\\_array](#) (int item, [circular\\_array \\*ca](#))
- void [destroy\\_circular\\_array](#) ([circular\\_array \\*ca](#))

#### 4.8.1 Detailed Description

Provides the implementation of a circular array data-structure.

#### 4.8.2 Typedef Documentation

4.8.2.1 typedef struct [circular\\_array](#) [circular\\_array](#)

#### 4.8.3 Function Documentation

4.8.3.1 int [add\\_to\\_circular\\_array](#) ( int *item*, [circular\\_array \\* ca](#) )

4.8.3.2 void [destroy\\_circular\\_array](#) ( [circular\\_array \\* ca](#) )

### 4.8.3.3 circular\_array\* new\_circular\_array ( int size )

## 4.9 combinations.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "combinations.h"
#include "llist.h"
#include "utils.h"
#include "tree.h"
#include "globals_comb.h"
```

### Functions

- int [comb](#) (llist \*\*graphl, int c, int l, int k, set \*alpha, tree \*b)
- void [collect\\_solutions](#) (int \*counts, int size, llist \*branches, llist \*solutions\_list)
- void [collect\\_solutions\\_with\\_all\\_combs](#) (int \*counts, int size, llist \*branches, llist \*solutions\_list)
- void [print\\_combinations](#) (llist \*combs)
- void [print\\_combination](#) (llist \*comb)
- int [check\\_constraint](#) (llist \*combR, llist \*combL)
- int [dac\\_check\\_join](#) (int var, int l, llist\_node \*\*assigned\_values, llist \*\*domain\_lists, llist \*\*undo\_list, int \*cc, llist \*remaining\_vars)
- int [mac\\_check\\_join](#) (int var, int l, llist\_node \*\*assigned\_values, llist \*\*domain\_lists, llist \*\*undo\_list, int \*cc, llist \*remaining\_vars)
- int [forward\\_check\\_join](#) (int var, int l, llist\_node \*\*assigned\_values, llist \*\*domain\_lists, llist \*\*undo\_list, int \*cc, llist \*remaining\_vars)
- int [assign\\_next\\_join](#) (int x, llist\_node \*\*assigned\_values, llist \*\*domain\_lists, llist \*\*assigned\_current\_domains)
- void [undo\\_filtering\\_join](#) (int l, llist \*\*undo\_list, llist \*\*domain\_lists)
- void [backtrack\\_search\\_join](#) (int l, llist \*\*\*solutions, llist \*solutions\_list)
- int [choose\\_var\\_join](#) (llist \*remaining\_vars, llist \*\*domain\_lists)
- int [backtrack\\_search\\_join\\_rec](#) (int l, llist\_node \*\*assigned\_values, llist \*\*assigned\_current\_domains, llist \*remaining\_vars, llist \*\*\*solutions, llist \*solutions\_list, llist \*\*domain\_lists)
- llist \*\* [get\\_solutions\\_of\\_size](#) (tree \*n, int k)
- void [combine\\_check\\_solution](#) (int \*counts, int size, void \*arg1, void \*arg2)
- void [count\\_base\\_k\\_function](#) (int \*xk, int i, int l, int k, llist \*children, llist \*solution\_list)
- void [memo\\_count\\_base\\_k\\_function](#) (int \*xk, int i, int l, int k, llist \*children, llist \*solution\_list)
- void [count\\_base\\_k](#) (int k, int l, llist \*children, llist \*solution\_list)
- void [count\\_base\\_kv](#) (int \*kv, int l, void function(int \*x, int y, void \*z, void \*w), void \*arg1, void \*arg2)
- void [markup\\_node\\_tree\\_recursive](#) (tree \*b, int \*gcolors)
- void [markup\\_node\\_tree](#) (tree \*b, int \*gcolors, int vertices)
- void [print\\_node\\_tree](#) (tree \*b, int \*gcolors, int t)
- int [get\\_combinations](#) (llist \*\*graphl, int vertices, int c, llist \*combs)
- int [hide\\_get\\_combinations](#) (llist \*\*graphl, int vertices, int c, llist \*combs)
- int [get\\_combinations\\_with](#) (int n, llist \*\*graphl, int vertices, int \*gcolors, int c, llist \*combs)
- void [multipliers](#) (int k, int d, int \*sizes\_d, int depth, tree \*mults, llist \*leafs)
- void [n\\_choose\\_i](#) (int n, int i, int l, tree \*chose, llist \*leafs)
- void [mark\\_cmap](#) (int c, llist \*\*marked)
- void [clear\\_cmarks](#) (llist \*marked)
- void [dfs\\_on\\_marked](#) (int n, llist \*\*graphl)
- int [check\\_connected\\_subgraph](#) (llist \*cgns, llist \*\*graphl)
- int [graph\\_connected](#) (llist \*\*graphl, int vertices)
- void [get\\_distance\\_k\\_nodes\\_rec](#) (llist \*\*graphl, int node, int k, int vertices, llist \*dirty, llist \*collected, int dfo)

- `llist * get_distance_k_nodes (llist **graphl, int node, int k, int vertices)`
- `int get_bf_combinations_from_subgraphs (llist **graphl, int c, int vertices)`
- `int get_bf_combinations_list_8 (llist **graphl, llist *vertices)`
- `int get_bf_combinations_list_7 (llist **graphl, llist *vertices)`
- `int get_bf_combinations_list_6 (llist **graphl, llist *vertices)`
- `int get_bf_combinations_list_5 (llist **graphl, llist *vertices)`
- `int get_bf_combinations_list_4 (llist **graphl, llist *vertices)`
- `int get_bf_combinations_list_3 (llist **graphl, llist *vertices)`
- `int get_bf_combinations_list_2 (llist **graphl, llist *vertices)`
- `int get_bf_combinations_8 (llist **graphl, int vertices)`
- `int get_bf_combinations_7 (llist **graphl, int vertices)`
- `int get_bf_combinations_6 (llist **graphl, int vertices)`
- `int get_bf_combinations_5 (llist **graphl, int vertices)`
- `int get_bf_combinations_4 (llist **graphl, int vertices)`
- `int get_bf_combinations_3 (llist **graphl, int vertices)`
- `int get_bf_combinations_2 (llist **graphl, int vertices)`

#### 4.9.1 Function Documentation

4.9.1.1 `int assign_next_join ( int x, llist_node ** assigned_values, llist ** domain_lists, llist ** assigned_current_domains )`

4.9.1.2 `void backtrack_search_join ( int l, llist *** solutions, llist * solutions_list )`

4.9.1.3 `int backtrack_search_join_rec ( int l, llist_node ** assigned_values, llist ** assigned_current_domains, llist * remaining_vars, llist *** solutions, llist * solutions_list, llist ** domain_lists )`

4.9.1.4 `int check_connected_subgraph ( llist * cgns, llist ** graphl )`

4.9.1.5 `int check_constraint ( llist * combR, llist * combL )`

4.9.1.6 `int choose_var_join ( llist * remaining_vars, llist ** domain_lists )`

4.9.1.7 `void clear_cmarks ( llist * marked )`

4.9.1.8 `void collect_solutions ( int * counts, int size, llist * branches, llist * solutions_list )`

4.9.1.9 `void collect_solutions_with_all_combs ( int * counts, int size, llist * branches, llist * solutions_list )`

4.9.1.10 `int comb ( llist ** graphl, int c, int l, int k, set * alpha, tree * b )`

4.9.1.11 `void combine_check_solution ( int * counts, int size, void * arg1, void * arg2 )`

4.9.1.12 `void count_base_k ( int k, int l, llist * children, llist * solution_list )`

4.9.1.13 `void count_base_k_function ( int * xk, int i, int l, int k, llist * children, llist * solution_list )`

4.9.1.14 `void count_base_kv ( int * kv, int l, void function(int *x, int y, void *z, void *w, void * arg1, void * arg2 )`

4.9.1.15 `int dac_check_join ( int var, int l, llist_node ** assigned_values, llist ** domain_lists, llist ** undo_list, int * cc, llist * remaining_vars )`

4.9.1.16 `void dfs_on_marked ( int n, llist ** graphl )`

- 4.9.1.17 `int forward_check_join ( int var, int l, llist_node ** assigned_values, llist ** domain_lists, llist ** undo_list, int * cc, llist * remaining_vars )`
- 4.9.1.18 `int get_bf_combinations_2 ( llist ** graphl, int vertices )`
- 4.9.1.19 `int get_bf_combinations_3 ( llist ** graphl, int vertices )`
- 4.9.1.20 `int get_bf_combinations_4 ( llist ** graphl, int vertices )`
- 4.9.1.21 `int get_bf_combinations_5 ( llist ** graphl, int vertices )`
- 4.9.1.22 `int get_bf_combinations_6 ( llist ** graphl, int vertices )`
- 4.9.1.23 `int get_bf_combinations_7 ( llist ** graphl, int vertices )`
- 4.9.1.24 `int get_bf_combinations_8 ( llist ** graphl, int vertices )`
- 4.9.1.25 `int get_bf_combinations_from_subgraphs ( llist ** graphl, int c, int vertices )`
- 4.9.1.26 `int get_bf_combinations_list_2 ( llist ** graphl, llist * vertices )`
- 4.9.1.27 `int get_bf_combinations_list_3 ( llist ** graphl, llist * vertices )`
- 4.9.1.28 `int get_bf_combinations_list_4 ( llist ** graphl, llist * vertices )`
- 4.9.1.29 `int get_bf_combinations_list_5 ( llist ** graphl, llist * vertices )`
- 4.9.1.30 `int get_bf_combinations_list_6 ( llist ** graphl, llist * vertices )`
- 4.9.1.31 `int get_bf_combinations_list_7 ( llist ** graphl, llist * vertices )`
- 4.9.1.32 `int get_bf_combinations_list_8 ( llist ** graphl, llist * vertices )`
- 4.9.1.33 `int get_combinations ( llist ** graphl, int vertices, int c, llist * combs )`
- 4.9.1.34 `int get_combinations_with ( int n, llist ** graphl, int vertices, int * gcolors, int c, llist * combs )`
- 4.9.1.35 `llist* get_distance_k_nodes ( llist ** graphl, int node, int k, int vertices )`
- 4.9.1.36 `void get_distance_k_nodes_rec ( llist ** graphl, int node, int k, int vertices, llist * dirty, llist * collected, int dfo )`
- 4.9.1.37 `llist** get_solutions_of_size ( tree * n, int k )`
- 4.9.1.38 `int graph_connected ( llist ** graphl, int vertices )`
- 4.9.1.39 `int hide_get_combinations ( llist ** graphl, int vertices, int c, llist * combs )`
- 4.9.1.40 `int mac_check_join ( int var, int l, llist_node ** assigned_values, llist ** domain_lists, llist ** undo_list, int * cc, llist * remaining_vars )`
- 4.9.1.41 `void mark_cmap ( int c, llist ** marked )`
- 4.9.1.42 `void markup_node_tree ( tree * b, int * gcolors, int vertices )`
- 4.9.1.43 `void markup_node_tree_recursive ( tree * b, int * gcolors )`

- 4.9.1.44 void `memo_count_base_k_function` ( int \* *xk*, int *i*, int *l*, int *k*, *l*list \* *children*, *l*list \* *solution\_list* )
- 4.9.1.45 void `multipliers` ( int *k*, int *d*, int \* *sizes\_d*, int *depth*, tree \* *mults*, *l*list \* *leafs* )
- 4.9.1.46 void `n_choose_i` ( int *n*, int *i*, int *l*, tree \* *chose*, *l*list \* *leafs* )
- 4.9.1.47 void `print_combination` ( *l*list \* *comb* )
- 4.9.1.48 void `print_combinations` ( *l*list \* *combs* )
- 4.9.1.49 void `print_node_tree` ( tree \* *b*, int \* *gcolors*, int *t* )
- 4.9.1.50 void `undo_filtering_join` ( int *l*, *l*list \*\* *undo\_list*, *l*list \*\* *domain\_lists* )

## 4.10 combinations.h File Reference

```
#include "llist.h"
#include "tree.h"
#include "set.h"
```

### Classes

- struct [solution\\_structure](#)

### Typedefs

- typedef struct [solution\\_structure](#) [solution\\_structure](#)

### Functions

- int [compare\\_nodes](#) (void \**va*, void \**vb*)
- int [comb](#) (*l*list \*\**graphl*, int *c*, int *l*, int *k*, [set](#) \**alpha*, tree \**b*)
- void [collect\\_solutions](#) (int \**counts*, int *size*, *l*list \**branches*, *l*list \**solutions\_list*)
- *l*list \*\* [get\\_solutions\\_of\\_size](#) (tree \**n*, int *k*)
- void [combine\\_check\\_solution](#) (int \**counts*, int *size*, void \**arg1*, void \**arg2*)
- void [count\\_base\\_k](#) (int *k*, int *l*, *l*list \**children*, *l*list \**solution\_list*)
- void [count\\_base\\_kv](#) (int \**kv*, int *l*, void function(int \**x*, int *y*, void \**z*, void \**w*), void \**arg1*, void \**arg2*)
- int [get\\_combinations](#) (*l*list \*\**graphl*, int *vertices*, int *c*, *l*list \**combs*)
- int [get\\_combinations\\_with](#) (int *n*, *l*list \*\**graphl*, int *vertices*, int \**gcolors*, int *c*, *l*list \**combs*)
- void [print\\_node\\_tree](#) (tree \**b*, int \**gcolors*, int *t*)
- void [markup\\_node\\_tree\\_recursive](#) (tree \**b*, int \**gcolors*)
- void [markup\\_node\\_tree](#) (tree \**b*, int \**gcolors*, int *vertices*)
- void [multipliers](#) (int *k*, int *d*, int \**sizes\_d*, int *depth*, tree \**mults*, *l*list \**leafs*)
- void [collect\\_solutions\\_with\\_all\\_combs](#) (int \**counts*, int *size*, *l*list \**branches*, *l*list \**solutions\_list*)
- void [count\\_base\\_k\\_function](#) (int \**xk*, int *i*, int *l*, int *k*, *l*list \**children*, *l*list \**solution\_list*)
- void [memo\\_count\\_base\\_k\\_function](#) (int \**xk*, int *i*, int *l*, int *k*, *l*list \**children*, *l*list \**solution\_list*)
- void [xxcount\\_base\\_k\\_function](#) (int \**xk*, int *i*, int *l*, int *k*, void *arg00*(), void \**arg01*, void \**arg02*)
- void [n\\_choose\\_i](#) (int *n*, int *i*, int *l*, tree \**chose*, *l*list \**leafs*)
- int [hide\\_get\\_combinations](#) (*l*list \*\**graphl*, int *vertices*, int *c*, *l*list \**combs*)
- int [get\\_bf\\_combinations\\_8](#) (*l*list \*\**graphl*, int *vertices*)
- int [get\\_bf\\_combinations\\_7](#) (*l*list \*\**graphl*, int *vertices*)
- int [get\\_bf\\_combinations\\_6](#) (*l*list \*\**graphl*, int *vertices*)

- int [get\\_bf\\_combinations\\_5](#) (llist \*\*graphl, int vertices)
- int [get\\_bf\\_combinations\\_4](#) (llist \*\*graphl, int vertices)
- int [get\\_bf\\_combinations\\_3](#) (llist \*\*graphl, int vertices)
- int [get\\_bf\\_combinations\\_2](#) (llist \*\*graphl, int vertices)
- void [mark\\_cmap](#) (int c, llist \*\*marked)
- void [clear\\_cmarks](#) (llist \*marked)
- void [dfs\\_on\\_marked](#) (int n, llist \*\*graphl)
- int [check\\_connected\\_subgraph](#) (llist \*cgns, llist \*\*graphl)
- int [graph\\_connected](#) (llist \*\*graphl, int vertices)
- int [sac\\_check\\_join](#) (int var, int l, llist\_node \*\*assigned\_values, llist \*\*domain\_lists, llist \*\*undo\_list, int \*cc)
- int [dac\\_check\\_join](#) (int var, int l, llist\_node \*\*assigned\_values, llist \*\*domain\_lists, llist \*\*undo\_list, int \*cc, llist \*remaining\_vars)
- int [forward\\_check\\_join](#) (int var, int l, llist\_node \*\*assigned\_values, llist \*\*domain\_lists, llist \*\*undo\_list, int \*cc, llist \*remaining\_vars)
- int [assign\\_next\\_join](#) (int x, llist\_node \*\*assigned\_values, llist \*\*domain\_lists, llist \*\*assigned\_current\_domains)
- void [undo\\_filtering\\_join](#) (int l, llist \*\*undo\_list, llist \*\*domain\_lists)
- int [fc\\_join](#) (int l, llist \*\*\*solutions, llist \*solutions\_list, llist \*\*passed\_domain\_lists)
- int [mac\\_check\\_join](#) (int var, int l, llist\_node \*\*assigned\_values, llist \*\*domain\_lists, llist \*\*undo\_list, int \*cc, llist \*remaining\_vars)
- int [check\\_constraint](#) (llist \*comba, llist \*combb)
- int [revise\\_combs](#) (int \*tuple, llist \*\*\*solutions, int \*\*\*last, char \*\*alive)
- int [apply\\_ac\\_join](#) (llist \*\*\*solutions, set \*queue)
- void [print\\_combinations](#) (llist \*combs)
- void [print\\_combination](#) (llist \*comb)
- int [choose\\_var\\_join](#) (llist \*remaining\_vars, llist \*\*domain\_lists)
- void [get\\_distance\\_k\\_nodes\\_rec](#) (llist \*\*graphl, int node, int k, int vertices, llist \*dirty, llist \*collected, int dfo)
- llist \* [get\\_distance\\_k\\_nodes](#) (llist \*\*graphl, int node, int k, int vertices)
- int [get\\_bf\\_combinations\\_from\\_subgraphs](#) (llist \*\*graphl, int c, int vertices)
- int [get\\_bf\\_combinations\\_list\\_8](#) (llist \*\*graphl, llist \*vertices)
- int [get\\_bf\\_combinations\\_list\\_7](#) (llist \*\*graphl, llist \*vertices)
- int [get\\_bf\\_combinations\\_list\\_6](#) (llist \*\*graphl, llist \*vertices)
- int [get\\_bf\\_combinations\\_list\\_5](#) (llist \*\*graphl, llist \*vertices)
- int [get\\_bf\\_combinations\\_list\\_4](#) (llist \*\*graphl, llist \*vertices)
- int [get\\_bf\\_combinations\\_list\\_3](#) (llist \*\*graphl, llist \*vertices)
- int [get\\_bf\\_combinations\\_list\\_2](#) (llist \*\*graphl, llist \*vertices)
- void [backtrack\\_search\\_join](#) (int l, llist \*\*\*solutions, llist \*solutions\_list)
- int [backtrack\\_search\\_join\\_rec](#) (int l, llist\_node \*\*assigned\_values, llist \*\*assigned\_current\_domains, llist \*remaining\_vars, llist \*\*\*solutions, llist \*solutions\_list, llist \*\*domain\_lists)

#### 4.10.1 Detailed Description

Provides the implementations for several algorithms for computing connected combinations of nodes.

#### 4.10.2 Typedef Documentation

##### 4.10.2.1 typedef struct `solution_structure` `solution_structure`

#### 4.10.3 Function Documentation

##### 4.10.3.1 int `apply_ac_join` ( llist \*\*\* `solutions`, set \* `queue` )

- 4.10.3.2 int assign\_next\_join ( int *x*, llist\_node \*\* *assigned\_values*, llist \*\* *domain\_lists*, llist \*\* *assigned\_current\_domains* )
- 4.10.3.3 void backtrack\_search\_join ( int *l*, llist \*\*\* *solutions*, llist \* *solutions\_list* )
- 4.10.3.4 int backtrack\_search\_join\_rec ( int *l*, llist\_node \*\* *assigned\_values*, llist \*\* *assigned\_current\_domains*, llist \* *remaining\_vars*, llist \*\*\* *solutions*, llist \* *solutions\_list*, llist \*\* *domain\_lists* )
- 4.10.3.5 int check\_connected\_subgraph ( llist \* *cgns*, llist \*\* *graphl* )
- 4.10.3.6 int check\_constraint ( llist \* *comba*, llist \* *combb* )
- 4.10.3.7 int choose\_var\_join ( llist \* *remaining\_vars*, llist \*\* *domain\_lists* )
- 4.10.3.8 void clear\_cmarks ( llist \* *marked* )
- 4.10.3.9 void collect\_solutions ( int \* *counts*, int *size*, llist \* *branches*, llist \* *solutions\_list* )
- 4.10.3.10 void collect\_solutions\_with\_all\_combs ( int \* *counts*, int *size*, llist \* *branches*, llist \* *solutions\_list* )
- 4.10.3.11 int comb ( llist \*\* *graphl*, int *c*, int *l*, int *k*, set \* *alpha*, tree \* *b* )
- 4.10.3.12 void combine\_check\_solution ( int \* *counts*, int *size*, void \* *arg1*, void \* *arg2* )
- 4.10.3.13 int compare\_nodes ( void \* *va*, void \* *vb* )
- 4.10.3.14 void count\_base\_k ( int *k*, int *l*, llist \* *children*, llist \* *solution\_list* )
- 4.10.3.15 void count\_base\_k\_function ( int \* *xk*, int *i*, int *l*, int *k*, llist \* *children*, llist \* *solution\_list* )
- 4.10.3.16 void count\_base\_kv ( int \* *kv*, int *l*, void function(int \**x*, int *y*, void \**z*, void \**w*, void \* *arg1*, void \* *arg2* )
- 4.10.3.17 int dac\_check\_join ( int *var*, int *l*, llist\_node \*\* *assigned\_values*, llist \*\* *domain\_lists*, llist \*\* *undo\_list*, int \* *cc*, llist \* *remaining\_vars* )
- 4.10.3.18 void dfs\_on\_marked ( int *n*, llist \*\* *graphl* )
- 4.10.3.19 int fc\_join ( int *l*, llist \*\*\* *solutions*, llist \* *solutions\_list*, llist \*\* *passed\_domain\_lists* )
- 4.10.3.20 int forward\_check\_join ( int *var*, int *l*, llist\_node \*\* *assigned\_values*, llist \*\* *domain\_lists*, llist \*\* *undo\_list*, int \* *cc*, llist \* *remaining\_vars* )
- 4.10.3.21 int get\_bf\_combinations\_2 ( llist \*\* *graphl*, int *vertices* )
- 4.10.3.22 int get\_bf\_combinations\_3 ( llist \*\* *graphl*, int *vertices* )
- 4.10.3.23 int get\_bf\_combinations\_4 ( llist \*\* *graphl*, int *vertices* )
- 4.10.3.24 int get\_bf\_combinations\_5 ( llist \*\* *graphl*, int *vertices* )
- 4.10.3.25 int get\_bf\_combinations\_6 ( llist \*\* *graphl*, int *vertices* )
- 4.10.3.26 int get\_bf\_combinations\_7 ( llist \*\* *graphl*, int *vertices* )
- 4.10.3.27 int get\_bf\_combinations\_8 ( llist \*\* *graphl*, int *vertices* )

- 4.10.3.28 int get\_bf\_combinations\_from\_subgraphs ( llist \*\* graphl, int c, int vertices )
- 4.10.3.29 int get\_bf\_combinations\_list\_2 ( llist \*\* graphl, llist \* vertices )
- 4.10.3.30 int get\_bf\_combinations\_list\_3 ( llist \*\* graphl, llist \* vertices )
- 4.10.3.31 int get\_bf\_combinations\_list\_4 ( llist \*\* graphl, llist \* vertices )
- 4.10.3.32 int get\_bf\_combinations\_list\_5 ( llist \*\* graphl, llist \* vertices )
- 4.10.3.33 int get\_bf\_combinations\_list\_6 ( llist \*\* graphl, llist \* vertices )
- 4.10.3.34 int get\_bf\_combinations\_list\_7 ( llist \*\* graphl, llist \* vertices )
- 4.10.3.35 int get\_bf\_combinations\_list\_8 ( llist \*\* graphl, llist \* vertices )
- 4.10.3.36 int get\_combinations ( llist \*\* graphl, int vertices, int c, llist \* combs )
- 4.10.3.37 int get\_combinations\_with ( int n, llist \*\* graphl, int vertices, int \* gcolors, int c, llist \* combs )
- 4.10.3.38 llist\* get\_distance\_k\_nodes ( llist \*\* graphl, int node, int k, int vertices )
- 4.10.3.39 void get\_distance\_k\_nodes\_rec ( llist \*\* graphl, int node, int k, int vertices, llist \* dirty, llist \* collected, int dfo )
- 4.10.3.40 llist\*\* get\_solutions\_of\_size ( tree \* n, int k )
- 4.10.3.41 int graph\_connected ( llist \*\* graphl, int vertices )
- 4.10.3.42 int hide\_get\_combinations ( llist \*\* graphl, int vertices, int c, llist \* combs )
- 4.10.3.43 int mac\_check\_join ( int var, int l, llist\_node \*\* assigned\_values, llist \*\* domain\_lists, llist \*\* undo\_list, int \* cc, llist \* remaining\_vars )
- 4.10.3.44 void mark\_cmap ( int c, llist \*\* marked )
- 4.10.3.45 void markup\_node\_tree ( tree \* b, int \* gcolors, int vertices )
- 4.10.3.46 void markup\_node\_tree\_recursive ( tree \* b, int \* gcolors )
- 4.10.3.47 void memo\_count\_base\_k\_function ( int \* xk, int i, int l, int k, llist \* children, llist \* solution\_list )
- 4.10.3.48 void multipliers ( int k, int d, int \* sizes\_d, int depth, tree \* mults, llist \* leafs )
- 4.10.3.49 void n\_choose\_i ( int n, int i, int l, tree \* chose, llist \* leafs )
- 4.10.3.50 void print\_combination ( llist \* comb )
- 4.10.3.51 void print\_combinations ( llist \* combs )
- 4.10.3.52 void print\_node\_tree ( tree \* b, int \* gcolors, int t )
- 4.10.3.53 int revise\_combs ( int \* tuple, llist \*\*\* solutions, int \*\*\* last, char \*\* alive )
- 4.10.3.54 int sac\_check\_join ( int var, int l, llist\_node \*\* assigned\_values, llist \*\* domain\_lists, llist \*\* undo\_list, int \* cc )



4.10.3.55 void `undo_filtering_join` ( int *l*, llist \*\* *undo\_list*, llist \*\* *domain\_lists* )

4.10.3.56 void `xxcount_base_k_function` ( int \* *xk*, int *i*, int *l*, int *k*, void *arg00()*, void \* *arg01*, void \* *arg02* )

## 4.11 constraints.c File Reference

```
#include "constraints.h"
#include "variables.h"
#include "utils.h"
#include "query_graph.h"
#include "decomposition.h"
```

### Functions

- `constraint * new_constraint` (char \**name*, int *id*, int \**vars*, int *fill*)
- `constraints * init_constraints` (int *count*)
- `constraint * add_constraint` (char \**name*, int *function*, `constraints *const`s, int *arity*, int *const\_id*, int *parameter\_count*, tuple \**tuples*, int *ac3*)
- void `add_variable_to_last_constraint` (`constraints *const`s, `variables *vars`, int *var\_id*, int *ac3*)
- void `add_parameter_to_last_constraint` (`constraints *const`s, void \**value*, int *value\_type*, int *pos*)
- void `reset_ac3_1_last` (`constraints *const`s, `variables *vars`)
- llist \* `split_constraints_into_connected_components` (`constraints *const`s, int *set\_var\_constraints*)
- llist \* `split_constraints_into_connected_components_list` (llist \**const*s, int *set\_var\_constraints*)
- void `split_constraints_into_connected_components_rec` (`constraint *cons`, char \**map*, llist \**maplist*)
- void `add_htables_to_constraints` (void)
- void `printClusterGraphNeato` (llist \**constraints*)
- `node_graph * dual_node_graph_from_constraints` (llist \**constraints*)
- `constraint_tree * get_constraint_tree` (llist \**constraints*)
- `constraint_tree * new_constraint_tree` ()
- `constraint_tree_node * new_constraint_tree_node` (llist \**clique*)
- `constraint_tree * get_dual_join_tree` (llist \**cliques*)
- void `print_consistant_tree` (`constraint_tree *ct`)
- void `constraints_make_this_root` (`constraint_tree_node *newroot`)
- void `constraints_swap_parent_child` (`constraint_tree_node *child`)
- int `get_unassigned_htables` (`constraint_tree_node *node`)
- void `print_number_of_deleted_tuples_values` (llist \**usedhts\_list*)
- void `add_constraint_to_variable` (`variable *var`, `constraint *constr`)

### Variables

- int `constraint_ids` = 0

#### 4.11.1 Function Documentation

4.11.1.1 `constraint* add_constraint` ( char \* *name*, int *function*, `constraints * const`s, int *arity*, int *const\_id*, int *parameter\_count*, tuple \* *tuples*, int *ac3* )

4.11.1.2 void `add_constraint_to_variable` ( `variable * var`, `constraint * constr` )

4.11.1.3 void `add_htables_to_constraints` ( void )

4.11.1.4 void `add_parameter_to_last_constraint` ( `constraints * const`s, void \* *value*, int *value\_type*, int *pos* )

- 4.11.1.5 void `add_variable_to_last_constraint` ( `constraints * consts`, `variables * vars`, `int var_id`, `int ac3` )
- 4.11.1.6 void `constraints_make_this_root` ( `constraint_tree_node * newroot` )
- 4.11.1.7 void `constraints_swap_parent_child` ( `constraint_tree_node * child` )
- 4.11.1.8 `node_graph*` `dual_node_graph_from_constraints` ( `llist * constraints` ) [read]
- 4.11.1.9 `constraint_tree*` `get_constraint_tree` ( `llist * constraints` )
- 4.11.1.10 `constraint_tree*` `get_dual_join_tree` ( `llist * cliques` )
- 4.11.1.11 int `get_unassigned_htables` ( `constraint_tree_node * node` )
- 4.11.1.12 `constraints*` `init_constraints` ( `int count` )
- 4.11.1.13 `constraint*` `new_constraint` ( `char * name`, `int id`, `int * vars`, `int fill` )
- 4.11.1.14 `constraint_tree*` `new_constraint_tree` ( `void` )
- 4.11.1.15 `constraint_tree_node*` `new_constraint_tree_node` ( `llist * clique` )
- 4.11.1.16 void `print_consistant_tree` ( `constraint_tree * ct` )
- 4.11.1.17 void `print_number_of_deleted_tuples_values` ( `llist * usedhts_list` )
- 4.11.1.18 void `printClusterGraphNeato` ( `llist * constraints` )
- 4.11.1.19 void `reset_ac3_1_last` ( `constraints * consts`, `variables * vars` )
- 4.11.1.20 `llist*` `split_constraints_into_connected_components` ( `constraints * consts`, `int set_var_constraints` )
- 4.11.1.21 `llist*` `split_constraints_into_connected_components_list` ( `llist * consts`, `int set_var_constraints` )
- 4.11.1.22 void `split_constraints_into_connected_components_rec` ( `constraint * cons`, `char * map`, `llist * maplist` )

## 4.11.2 Variable Documentation

- 4.11.2.1 int `constraint_ids` = 0

## 4.12 constraints.h File Reference

```
#include <stdlib.h>
#include "variables.h"
#include "relations.h"
```

### Classes

- struct [parameter](#)
- struct [constraint](#)
- struct [constraints](#)
- struct [constraint\\_tree](#)
- struct [constraint\\_tree\\_node](#)

## Typedefs

- typedef struct [parameter](#) [parameter](#)
- typedef struct [constraint](#) [constraint](#)
- typedef struct [constraints](#) [constraints](#)
- typedef struct [constraint\\_tree](#) [constraint\\_tree](#)
- typedef struct [constraint\\_tree\\_node](#) [constraint\\_tree\\_node](#)

## Functions

- [constraint](#) \* [new\\_constraint](#) (char \*name, int id, int \*vars, int fill)
- [constraints](#) \* [init\\_constraints](#) (int count)
- [constraint](#) \* [add\\_constraint](#) (char \*name, int function, [constraints](#) \*consts, int arity, int const\_id, int parameter\_count, tuple \*tuples, int ac3)
- void [add\\_variable\\_to\\_last\\_constraint](#) ([constraints](#) \*consts, [variables](#) \*vars, int [variable](#), int ac3)
- void [add\\_parameter\\_to\\_last\\_constraint](#) ([constraints](#) \*consts, void \*value, int type, int pos)
- void [reset\\_ac3\\_1\\_last](#) ([constraints](#) \*consts, [variables](#) \*vars)
- [llist](#) \* [split\\_constraints\\_into\\_connected\\_components](#) ([constraints](#) \*consts, int [set\\_variable\\_hashtable](#))
- [llist](#) \* [split\\_constraints\\_into\\_connected\\_components\\_list](#) ([llist](#) \*consts, int [set\\_variable\\_hashtable](#))
- void [split\\_constraints\\_into\\_connected\\_components\\_rec](#) ([constraint](#) \*cons, char \*map, [llist](#) \*maplist)
- void [add\\_htables\\_to\\_constraints](#) (void)
- void [printClusterGraphNeato](#) ([llist](#) \*constraints)
- struct [node\\_graph](#) \* [dual\\_node\\_graph\\_from\\_constraints](#) ([llist](#) \*constraints)
- [constraint\\_tree](#) \* [get\\_constraint\\_tree](#) ([llist](#) \*constraints)
- [constraint\\_tree](#) \* [get\\_dual\\_join\\_tree](#) ([llist](#) \*cliques)
- [constraint\\_tree](#) \* [new\\_constraint\\_tree](#) (void)
- [constraint\\_tree\\_node](#) \* [new\\_constraint\\_tree\\_node](#) ([llist](#) \*clique)
- void [print\\_consistant\\_tree](#) ([constraint\\_tree](#) \*ct)
- void [constraints\\_swap\\_parent\\_child](#) ([constraint\\_tree\\_node](#) \*child)
- void [constraints\\_make\\_this\\_root](#) ([constraint\\_tree\\_node](#) \*newroot)
- int [get\\_unassigned\\_htables](#) ([constraint\\_tree\\_node](#) \*node)
- void [print\\_number\\_of\\_deleted\\_tuples\\_values](#) ([llist](#) \*usedhts\_list)
- void [add\\_constraint\\_to\\_variable](#) ([variable](#) \*var, [constraint](#) \*constraint)

### 4.12.1 Detailed Description

Provides the data structures for representing the CSP constraints and methods for constructing and manipulating them.

### 4.12.2 Typedef Documentation

4.12.2.1 typedef struct [constraint](#) [constraint](#)

4.12.2.2 typedef struct [constraint\\_tree](#) [constraint\\_tree](#)

4.12.2.3 typedef struct [constraint\\_tree\\_node](#) [constraint\\_tree\\_node](#)

4.12.2.4 typedef struct [constraints](#) [constraints](#)

4.12.2.5 typedef struct [parameter](#) [parameter](#)

### 4.12.3 Function Documentation

- 4.12.3.1 `constraint*` `add_constraint` ( `char * name`, `int function`, `constraints * consts`, `int arity`, `int const_id`, `int parameter_count`, `tuple * tuples`, `int ac3` )
- 4.12.3.2 `void` `add_constraint_to_variable` ( `variable * var`, `constraint * constraint` )
- 4.12.3.3 `void` `add_htables_to_constraints` ( `void` )
- 4.12.3.4 `void` `add_parameter_to_last_constraint` ( `constraints * consts`, `void * value`, `int type`, `int pos` )
- 4.12.3.5 `void` `add_variable_to_last_constraint` ( `constraints * consts`, `variables * vars`, `int variable`, `int ac3` )
- 4.12.3.6 `void` `constraints_make_this_root` ( `constraint_tree_node * newroot` )
- 4.12.3.7 `void` `constraints_swap_parent_child` ( `constraint_tree_node * child` )
- 4.12.3.8 `struct node_graph*` `dual_node_graph_from_constraints` ( `llist * constraints` ) [read]
- 4.12.3.9 `constraint_tree*` `get_constraint_tree` ( `llist * constraints` )
- 4.12.3.10 `constraint_tree*` `get_dual_join_tree` ( `llist * cliques` )
- 4.12.3.11 `int` `get_unassigned_htables` ( `constraint_tree_node * node` )
- 4.12.3.12 `constraints*` `init_constraints` ( `int count` )
- 4.12.3.13 `constraint*` `new_constraint` ( `char * name`, `int id`, `int * vars`, `int fill` )
- 4.12.3.14 `constraint_tree*` `new_constraint_tree` ( `void` )
- 4.12.3.15 `constraint_tree_node*` `new_constraint_tree_node` ( `llist * clique` )
- 4.12.3.16 `void` `print_consistant_tree` ( `constraint_tree * ct` )
- 4.12.3.17 `void` `print_number_of_deleted_tuples_values` ( `llist * usedhts_list` )
- 4.12.3.18 `void` `printClusterGraphNeato` ( `llist * constraints` )
- 4.12.3.19 `void` `reset_ac3_1_last` ( `constraints * consts`, `variables * vars` )
- 4.12.3.20 `llist*` `split_constraints_into_connected_components` ( `constraints * consts`, `int set_variable_hashtable` )
- 4.12.3.21 `llist*` `split_constraints_into_connected_components_list` ( `llist * consts`, `int set_variable_hashtable` )
- 4.12.3.22 `void` `split_constraints_into_connected_components_rec` ( `constraint * cons`, `char * map`, `llist * maplist` )

## 4.13 decomposition.c File Reference

```
#include "decomposition.h"
#include "query_graph.h"
#include "utils.h"
#include "combinations_wrapper.h"
#include "rs2.h"
#include "tree_decomp.h"
#include "rsprocedures.h"
#include "tree.h"
#include "active_tuple.h"
#include "splitter.h"
#include "minibucket.h"
#include "partition.h"
#include "decomposition_functions.h"
#include "hashtable_search.h"
#include "tables.h"
#include "sort.h"
```

### Functions

- void [set\\_dual\\_heap\\_position](#) (int pos, void \*body)
- void [set\\_dual\\_heap\\_value](#) (int val, void \*body)
- int [get\\_dual\\_heap\\_value](#) (void \*body)
- [constraint\\_graph\\_node \\*\\* get\\_dual\\_elimination\\_ordering](#) ([constraint\\_graph](#) \*cg)
- int [count\\_dual\\_fill\\_edges](#) (char \*\*matrix, [constraint\\_graph\\_node](#) \*node)
- void [add\\_dual\\_fill\\_edges](#) (char \*\*matrix, [constraint\\_graph\\_node](#) \*node, [heap](#) \*h, [constraint\\_graph](#) \*cg)
- [llist](#) \* [dual\\_cliques](#) ([constraint\\_graph\\_node \\*\\*ordering](#), [constraint\\_graph](#) \*cg)
- [node\\_graph](#) \* [new\\_node\\_graph](#) ([constraint\\_graph](#) \*cg)
- [node\\_graph](#) \* [new\\_node\\_graph\\_wpeo](#) ([constraint\\_graph](#) \*cg, [node\\_graph](#) \*old\_graph\_get\_node\_peo\_from, [llist](#) \*added\_edges)
- [node\\_graph](#) \* [new\\_node\\_graph\\_from\\_rels](#) ([llist](#) \*rels, [llist](#) \*vars, int \*varmap)
- void [add\\_fill\\_edges\\_to\\_node\\_graph](#) ([node\\_graph](#) \*ng, [llist](#) \*filledges)
- void [destroy\\_node\\_graph](#) ([node\\_graph](#) \*ng)
- void [set\\_heap\\_position](#) (int pos, void \*body)
- int [get\\_heap\\_value](#) (void \*body)
- void [print\\_node\\_graph](#) ([node\\_graph](#) \*ng)
- int \* [get\\_elimination\\_ordering](#) ([node\\_graph](#) \*ng, [llist](#) \*added\_edges)
- int [count\\_fill\\_edges](#) ([node\\_graph](#) \*ng, int var)
- void [add\\_fill\\_edges](#) ([node\\_graph](#) \*ng, [heap](#) \*h, int var, [llist](#) \*added\_edges)
- void [ng\\_set\\_heap\\_pos](#) (int pos, void \*body)
- void [ng\\_set\\_heap\\_value](#) (int val, void \*body)
- int [ng\\_get\\_heap\\_value](#) (void \*body)
- int \* [max\\_cardinality](#) ([node\\_graph](#) \*ng, int node)
- [llist](#) \* [cliques](#) (int \*ordering, [node\\_graph](#) \*ng, int \*size)
- int [get\\_common\\_rels\\_count\\_from\\_sets](#) ([set](#) \*va, [set](#) \*vb, [llist](#) \*clist)
- int [get\\_common\\_vars\\_count\\_from\\_sets](#) ([set](#) \*va, [set](#) \*vb, [llist](#) \*clist)
- void [add\\_decomposed\\_tree\\_child](#) ([decomposed\\_tree](#) \*child, [decomposed\\_tree](#) \*parent)
- [decomposed\\_tree](#) \* [new\\_decomposed\\_tree](#) ([llist](#) \*vl)
- void [set\\_leafs](#) ([decomposed\\_tree](#) \*root)
- [decomposed\\_tree \\*\\* get\\_cluster\\_array](#) ([decomposed\\_tree](#) \*root)
- void [make\\_graph](#) ([decomposed\\_tree](#) \*root, [llist](#) \*leafs, [decomposed\\_tree \\*\\*\\*c\\_array](#))
- void [make\\_combinations\\_for\\_cluster](#) ([decomposed\\_tree](#) \*cluster, [llist](#) \*combs)
- int [compare\\_combs](#) (void \*o1, void \*o2)

- `int get_comb_node_id (void *body, void *bla)`
- `int setup_combinations (decomposed_tree **c_array, int clusters)`
- `void add_separator_relations (decomposed_tree **c_array, int clusters)`
- `void add_separator_relations_for_cluster (decomposed_tree *current, int clusters, int clear_combdatas)`
- `decomposed_tree * build_decomposed_tree_vertices (llist *cliques, llist *tree_nodes)`
- `decomposed_tree * find_smallest_root (llist *nodes)`
- `decomposed_tree * find_balancing_root (llist *nodes)`
- `decomposed_tree * find_max_degree_node (llist *nodes)`
- `void make_this_root (decomposed_tree *newroot)`
- `void swap_parent_child (decomposed_tree *child, decomposed_tree *parent)`
- `void merge_tree_nodes (decomposed_tree *root)`
- `void reid_decomposed_tree_nodes (decomposed_tree *root, int *id)`
- `void merge_tree_nodes_rec (decomposed_tree *treen)`
- `void merge_with_parent (decomposed_tree *treen)`
- `void print_decomposed_tree_node (decomposed_tree *tree, int tab, decomposition_info *di)`
- `void print_decomposed_tree_node_dot (decomposed_tree *tree, decomposition_info *di, FILE *f)`
- `void print_decomposed_tree_node_summary_dot (decomposed_tree *tree, decomposition_info *di, FILE *f)`
- `void print_decomposed_tree_cluster_dot (decomposed_tree *tree)`
- `void print_decomposed_tree_cluster (decomposed_tree *tree)`
- `decomposition_info * new_di (void)`
- `void print_decomposed_tree (decomposed_tree *tree, int tab, FILE *f, int summary, int print_screen)`
- `void print_decomposed_tree_rec (decomposed_tree *tree, int tab, int summary, decomposition_info *di, FILE *f)`
- `void set_uncovered_variables (decomposed_tree *root)`
- `void report_separator_quality (decomposed_tree *child)`
- `void print_filtering_report (constraints *cons)`
- `void check_all_constraints_added (constraints *cons, FILE *f)`
- `decomposed_tree * get_decomposed_tree_from_library (constraint_graph *cg, llist *tree_nodes, int tree_only_no_cg)`
- `decomposed_tree * get_decomposed_tree_from_library_rec (tree_node *tree, constraint_graph *cg, llist *tree_nodes, int tree_only_no_cg)`
- `decomposed_tree * get_dual_decomposed_tree_from_library (constraint_graph *cg, llist *tree_nodes, int tree_only_no_cg)`
- `decomposed_tree * get_dual_decomposed_tree_from_library_rec (tree_node *tree, constraint_graph *cg, llist *tree_nodes, int tree_only_no_cg)`
- `int label_levels (decomposed_tree *root)`
- `int add_affiliate_rels (llist *tree_nodes)`
- `void append_to_comb_head (htable *ht, llist *comb)`
- `void append_to_comb (htable *ht, llist *comb)`
- `void hide_comb_datas (decomposed_tree *node)`
- `void un_hide_comb_datas (decomposed_tree *node)`
- `void cluster_pre_process (decomposed_tree *treen, int copy_tables)`
- `void cluster_pre_process2 (decomposed_tree *treen, int copy_tables)`
- `int set_subtree_size (decomposed_tree *root)`
- `void label_depth (decomposed_tree *root)`
- `void clear_combdatas_from_relations (decomposed_tree *root)`
- `void clear_combdatas_from_relations_only_root (decomposed_tree *root, int onlyroot)`
- `void split_trees (decomposed_tree *tree, llist *trees)`
- `void print_decomposition_tree_tables_h (decomposed_tree *treen)`
- `void print_decomposition_tree_tables (decomposed_tree *treen)`
- `void normalized_projected_rels_on_c_vars (decomposed_tree *treen)`
- `void set_pre_post (decomposed_tree *treen, int *id, decomposed_tree **treens)`
- `int in_subtree (decomposed_tree *root, decomposed_tree *node)`
- `void generate_xml (decomposed_tree *tree, int idp, int idp2, int idp3)`
- `void report_tree_separators (decomposed_tree *treen)`
- `void report_tree_separators2 (decomposed_tree *treen)`
- `void print_tree_report (decomposed_tree *tree, double *maxratio, int *maxcv, int *maxsv)`

### 4.13.1 Function Documentation

- 4.13.1.1 `int add_affiliate_rels ( llist * tree_nodes )`
- 4.13.1.2 `void add_decomposed_tree_child ( decomposed_tree * child, decomposed_tree * parent )`
- 4.13.1.3 `void add_dual_fill_edges ( char ** matrix, constraint_graph_node * node, heap * h, constraint_graph * cg )`
- 4.13.1.4 `void add_fill_edges ( node_graph * ng, heap * h, int var, llist * added_edges )`
- 4.13.1.5 `void add_fill_edges_to_node_graph ( node_graph * ng, llist * filledges )`
- 4.13.1.6 `void add_separator_relations ( decomposed_tree ** c_array, int clusters )`
- 4.13.1.7 `void add_separator_relations_for_cluster ( decomposed_tree * current, int clusters, int clear_combdatas )`
- 4.13.1.8 `void append_to_comb ( htable * ht, llist * comb )`
- 4.13.1.9 `void append_to_comb_head ( htable * ht, llist * comb )`
- 4.13.1.10 `decomposed_tree* build_decomposed_tree_vertices ( llist * cliques, llist * tree_nodes )`
- 4.13.1.11 `void check_all_constraints_added ( constraints * cons, FILE * f )`
- 4.13.1.12 `void clear_combdatas_from_relations ( decomposed_tree * root )`
- 4.13.1.13 `void clear_combdatas_from_relations_only_root ( decomposed_tree * root, int onlyroot )`
- 4.13.1.14 `llist* cliques ( int * ordering, node_graph * ng, int * size )`
- 4.13.1.15 `void cluster_pre_process ( decomposed_tree * treen, int copy_tables )`
- 4.13.1.16 `void cluster_pre_process2 ( decomposed_tree * treen, int copy_tables )`
- 4.13.1.17 `int compare_combs ( void * o1, void * o2 )`
- 4.13.1.18 `int count_dual_fill_edges ( char ** matrix, constraint_graph_node * node )`
- 4.13.1.19 `int count_fill_edges ( node_graph * ng, int var )`
- 4.13.1.20 `void destroy_node_graph ( node_graph * ng )`
- 4.13.1.21 `llist* dual_cliques ( constraint_graph_node ** ordering, constraint_graph * cg )`
- 4.13.1.22 `decomposed_tree* find_balancing_root ( llist * nodes )`
- 4.13.1.23 `decomposed_tree* find_max_degree_node ( llist * nodes )`
- 4.13.1.24 `decomposed_tree* find_smallest_root ( llist * nodes )`
- 4.13.1.25 `void generate_xml ( decomposed_tree * tree, int idp, int idp2, int idp3 )`
- 4.13.1.26 `decomposed_tree** get_cluster_array ( decomposed_tree * root )`
- 4.13.1.27 `int get_comb_node_id ( void * body, void * bla )`

- 4.13.1.28 `int get_common_rels_count_from_sets ( set * va, set * vb, llist * clist )`
- 4.13.1.29 `int get_common_vars_count_from_sets ( set * va, set * vb, llist * clist )`
- 4.13.1.30 `decomposed_tree* get_decomposed_tree_from_library ( constraint_graph * cg, llist * tree_nodes, int tree_only_no_cg )`
- 4.13.1.31 `decomposed_tree* get_decomposed_tree_from_library_rec ( tree_node * tree, constraint_graph * cg, llist * tree_nodes, int tree_only_no_cg )`
- 4.13.1.32 `decomposed_tree* get_dual_decomposed_tree_from_library ( constraint_graph * cg, llist * tree_nodes, int tree_only_no_cg )`
- 4.13.1.33 `decomposed_tree* get_dual_decomposed_tree_from_library_rec ( tree_node * tree, constraint_graph * cg, llist * tree_nodes, int tree_only_no_cg )`
- 4.13.1.34 `constraint_graph_node** get_dual_elimination_ordering ( constraint_graph * cg )`
- 4.13.1.35 `int get_dual_heap_value ( void * body )`
- 4.13.1.36 `int* get_elimination_ordering ( node_graph * ng, llist * added_edges )`
- 4.13.1.37 `int get_heap_value ( void * body )`
- 4.13.1.38 `void hide_comb_datas ( decomposed_tree * node )`
- 4.13.1.39 `int in_subtree ( decomposed_tree * root, decomposed_tree * node )`
- 4.13.1.40 `void label_depth ( decomposed_tree * root )`
- 4.13.1.41 `int label_levels ( decomposed_tree * root )`
- 4.13.1.42 `void make_combinations_for_cluster ( decomposed_tree * cluster, llist * combs )`
- 4.13.1.43 `void make_graph ( decomposed_tree * root, llist * leafs, decomposed_tree *** c_array )`
- 4.13.1.44 `void make_this_root ( decomposed_tree * newroot )`
- 4.13.1.45 `int* max_cardinality ( node_graph * ng, int node )`
- 4.13.1.46 `void merge_tree_nodes ( decomposed_tree * root )`
- 4.13.1.47 `void merge_tree_nodes_rec ( decomposed_tree * treen )`
- 4.13.1.48 `void merge_with_parent ( decomposed_tree * treen )`
- 4.13.1.49 `decomposed_tree* new_decomposed_tree ( llist * vl )`
- 4.13.1.50 `decomposition_info* new_di ( void )`
- 4.13.1.51 `node_graph* new_node_graph ( constraint_graph * cg )`
- 4.13.1.52 `node_graph* new_node_graph_from_rels ( llist * rels, llist * vars, int * varmap )`
- 4.13.1.53 `node_graph* new_node_graph_wpeo ( constraint_graph * cg, node_graph * old_graph_get_node_peo_from, llist * added_edges )`



- 4.13.1.54 `int ng_get_heap_value ( void * body )`
- 4.13.1.55 `void ng_set_heap_pos ( int pos, void * body )`
- 4.13.1.56 `void ng_set_heap_value ( int val, void * body )`
- 4.13.1.57 `void normalized_projected_rels_on_c_vars ( decomposed_tree * treen )`
- 4.13.1.58 `void print_decomposed_tree ( decomposed_tree * tree, int tab, FILE * f, int summary, int print_screen )`
- 4.13.1.59 `void print_decomposed_tree_cluster ( decomposed_tree * tree )`
- 4.13.1.60 `void print_decomposed_tree_cluster_dot ( decomposed_tree * tree )`
- 4.13.1.61 `void print_decomposed_tree_node ( decomposed_tree * tree, int tab, decomposition_info * di )`
- 4.13.1.62 `void print_decomposed_tree_node_dot ( decomposed_tree * tree, decomposition_info * di, FILE * f )`
- 4.13.1.63 `void print_decomposed_tree_node_summary_dot ( decomposed_tree * tree, decomposition_info * di, FILE * f )`
- 4.13.1.64 `void print_decomposed_tree_rec ( decomposed_tree * tree, int tab, int summary, decomposition_info * di, FILE * f )`
- 4.13.1.65 `void print_decomposition_tree_tables ( decomposed_tree * treen )`
- 4.13.1.66 `void print_decomposition_tree_tables_h ( decomposed_tree * treen )`
- 4.13.1.67 `void print_filtering_report ( constraints * cons )`
- 4.13.1.68 `void print_node_graph ( node_graph * ng )`
- 4.13.1.69 `void print_tree_report ( decomposed_tree * tree, double * maxratio, int * maxcv, int * maxsv )`
- 4.13.1.70 `void reid_decomposed_tree_nodes ( decomposed_tree * root, int * id )`
- 4.13.1.71 `void report_separator_quality ( decomposed_tree * child )`
- 4.13.1.72 `void report_tree_separators ( decomposed_tree * treen )`
- 4.13.1.73 `void report_tree_separators2 ( decomposed_tree * treen )`
- 4.13.1.74 `void set_dual_heap_position ( int pos, void * body )`
- 4.13.1.75 `void set_dual_heap_value ( int val, void * body )`
- 4.13.1.76 `void set_heap_position ( int pos, void * body )`
- 4.13.1.77 `void set_leafs ( decomposed_tree * root )`
- 4.13.1.78 `void set_pre_post ( decomposed_tree * treen, int * id, decomposed_tree ** treens )`
- 4.13.1.79 `int set_subtree_size ( decomposed_tree * root )`
- 4.13.1.80 `void set_uncovered_variables ( decomposed_tree * root )`

4.13.1.81 `int setup_combinations ( decomposed_tree ** c_array, int clusters )`

4.13.1.82 `void split_trees ( decomposed_tree * tree, llist * trees )`

4.13.1.83 `void swap_parent_child ( decomposed_tree * child, decomposed_tree * parent )`

4.13.1.84 `void un_hide_comb_datas ( decomposed_tree * node )`

## 4.14 decomposition.h File Reference

```
#include "l1ist.h"
#include "query_graph.h"
#include "heap.h"
#include "set.h"
#include "tree_decomp.h"
#include "rs.h"
#include "rb_tree.h"
```

### Classes

- struct [decomposition\\_info](#)
- struct [node\\_graph](#)
- struct [separator\\_info](#)
- struct [decomposed\\_tree](#)

### Typedefs

- typedef struct [decomposition\\_info](#) [decomposition\\_info](#)
- typedef struct [node\\_graph](#) [node\\_graph](#)
- typedef struct [separator\\_info](#) [separator\\_info](#)
- typedef struct [decomposed\\_tree](#) [decomposed\\_tree](#)

### Functions

- [constraint\\_graph\\_node \\*\\* get\\_dual\\_elimination\\_ordering \(constraint\\_graph \\*cg\)](#)
- [int count\\_dual\\_fill\\_edges \(char \\*\\*matrix, constraint\\_graph\\_node \\*node\)](#)
- [void set\\_dual\\_heap\\_position \(int pos, void \\*body\)](#)
- [void set\\_dual\\_heap\\_value \(int val, void \\*body\)](#)
- [int get\\_dual\\_heap\\_value \(void \\*body\)](#)
- [void add\\_dual\\_fill\\_edges \(char \\*\\*matrix, constraint\\_graph\\_node \\*node, heap \\*h, constraint\\_graph \\*cg\)](#)
- [l1ist \\* dual\\_cliques \(constraint\\_graph\\_node \\*\\*ordering, constraint\\_graph \\*cg\)](#)
- [void print\\_node\\_graph \(node\\_graph \\*ng\)](#)
- [node\\_graph \\* new\\_node\\_graph \(constraint\\_graph \\*cg\)](#)
- [node\\_graph \\* new\\_node\\_graph\\_wpeo \(constraint\\_graph \\*cg, node\\_graph \\*old\\_graph\\_get\\_node\\_peo\\_from, l1ist \\*added\\_edges\)](#)
- [void destroy\\_node\\_graph \(node\\_graph \\*ng\)](#)
- [int \\* get\\_elimination\\_ordering \(node\\_graph \\*ng, l1ist \\*added\\_edges\)](#)
- [int count\\_fill\\_edges \(node\\_graph \\*ng, int var\)](#)
- [void set\\_heap\\_position \(int pos, void \\*body\)](#)
- [int get\\_heap\\_value \(void \\*body\)](#)
- [void add\\_fill\\_edges \(node\\_graph \\*ng, heap \\*h, int var, l1ist \\*added\\_edges\)](#)
- [l1ist \\* cliques \(int \\*ordering, node\\_graph \\*ng, int \\*size\)](#)

- int [get\\_common\\_vars\\_count\\_from\\_sets](#) (set \*va, set \*vb, llist \*clist)
- int [get\\_common\\_rels\\_count\\_from\\_sets](#) (set \*va, set \*vb, llist \*clist)
- void [add\\_decomposed\\_tree\\_child](#) (decomposed\_tree \*child, decomposed\_tree \*parent)
- void [print\\_decomposed\\_tree\\_node\\_dot](#) (decomposed\_tree \*tree, decomposition\_info \*di, FILE \*f)
- void [print\\_decomposed\\_tree\\_node\\_summary\\_dot](#) (decomposed\_tree \*tree, decomposition\_info \*di, FILE \*f)
- void [print\\_decomposed\\_tree](#) (decomposed\_tree \*tree, int tab, FILE \*f, int summary, int print\_screen)
- void [print\\_decomposed\\_tree\\_rec](#) (decomposed\_tree \*tree, int tab, int summary, decomposition\_info \*di, FILE \*f)
- decomposed\_tree \* [build\\_decomposed\\_tree\\_vertices](#) (llist \*cliques, llist \*tree\_nodes)
- decomposed\_tree \* [new\\_decomposed\\_tree](#) (llist \*vl)
- void [print\\_decomposed\\_tree\\_node](#) (decomposed\_tree \*tree, int tab, decomposition\_info \*di)
- decomposition\_info \* [new\\_di](#) (void)
- void [print\\_filtering\\_report](#) (constraints \*cons)
- void [check\\_all\\_constraints\\_added](#) (constraints \*cons, FILE \*f)
- decomposed\_tree \* [find\\_max\\_degree\\_node](#) (llist \*nodes)
- void [make\\_this\\_root](#) (decomposed\_tree \*newroot)
- void [swap\\_parent\\_child](#) (decomposed\_tree \*child, decomposed\_tree \*parent)
- decomposed\_tree \* [get\\_decomposed\\_tree\\_from\\_library](#) (constraint\_graph \*cg, llist \*tree\_nodes, int tree\_only\_no\_cg)
- decomposed\_tree \* [get\\_decomposed\\_tree\\_from\\_library\\_rec](#) (tree\_node \*tree, constraint\_graph \*cg, llist \*tree\_nodes, int tree\_only\_no\_cg)
- void [set\\_uncovered\\_variables](#) (decomposed\_tree \*root)
- decomposed\_tree \* [find\\_balancing\\_root](#) (llist \*nodes)
- void [init\\_cluster\\_queue](#) (decomposed\_tree \*tree)
- int [label\\_levels](#) (decomposed\_tree \*root)
- int [add\\_affiliate\\_rels](#) (llist \*tree\_nodes)
- void [append\\_to\\_comb](#) (htable \*ht, llist \*comb)
- void [hide\\_comb\\_datas](#) (decomposed\_tree \*node)
- void [un\\_hide\\_comb\\_datas](#) (decomposed\_tree \*node)
- int [filter\\_tuples\\_for\\_var\\_val](#) (variable \*var, value \*val)
- void [ng\\_set\\_heap\\_pos](#) (int pos, void \*body)
- void [ng\\_set\\_heap\\_value](#) (int val, void \*body)
- int [ng\\_get\\_heap\\_value](#) (void \*body)
- int \* [max\\_cardinality](#) (node\_graph \*ng, int node)
- void [append\\_to\\_comb\\_head](#) (htable \*ht, llist \*comb)
- void [print\\_decomposed\\_tree\\_cluster\\_dot](#) (decomposed\_tree \*tree)
- void [print\\_decomposed\\_tree\\_cluster](#) (decomposed\_tree \*tree)
- decomposed\_tree \* [get\\_dual\\_decomposed\\_tree\\_from\\_library](#) (constraint\_graph \*cg, llist \*tree\_nodes, int tree\_only\_no\_cg)
- decomposed\_tree \* [get\\_dual\\_decomposed\\_tree\\_from\\_library\\_rec](#) (tree\_node \*tree, constraint\_graph \*cg, llist \*tree\_nodes, int tree\_only\_no\_cg)
- void [cluster\\_pre\\_process](#) (decomposed\_tree \*treen, int copy\_tables)
- void [report\\_separator\\_quality](#) (decomposed\_tree \*child)
- node\_graph \* [new\\_node\\_graph\\_from\\_rels](#) (llist \*rels, llist \*vars, int \*varmap)
- void [add\\_fill\\_edges\\_to\\_node\\_graph](#) (node\_graph \*ng, llist \*filledges)
- void [label\\_depth](#) (decomposed\_tree \*root)
- void [clear\\_combdatas\\_from\\_relations](#) (decomposed\_tree \*root)
- void [clear\\_combdatas\\_from\\_relations\\_only\\_root](#) (decomposed\_tree \*root, int onlyroot)
- decomposed\_tree \* [find\\_smallest\\_root](#) (llist \*nodes)
- void [split\\_trees](#) (decomposed\_tree \*tree, llist \*trees)
- void [print\\_decomposition\\_tree\\_tables\\_h](#) (decomposed\_tree \*treen)
- void [print\\_decomposition\\_tree\\_tables](#) (decomposed\_tree \*treen)
- int [set\\_subtree\\_size](#) (decomposed\_tree \*root)
- void [normalized\\_projected\\_rels\\_on\\_c\\_vars](#) (decomposed\_tree \*treen)
- void [cluster\\_pre\\_process2](#) (decomposed\_tree \*treen, int copy\_tables)

- void `set_pre_post` (`decomposed_tree` \*tree, int \*id, `decomposed_tree` \*\*trees)
- int `in_subtree` (`decomposed_tree` \*root, `decomposed_tree` \*node)
- void `make_graph` (`decomposed_tree` \*root, `llist` \*leafs, `decomposed_tree` \*\*\*array)
- void `set_leafs` (`decomposed_tree` \*root)
- `decomposed_tree` \*\* `get_cluster_array` (`decomposed_tree` \*root)
- void `add_separator_relations` (`decomposed_tree` \*\*c\_array, int clusters)
- void `add_separator_relations_for_cluster` (`decomposed_tree` \*current, int clusters, int clear\_combdatas)
- int `setup_combinations` (`decomposed_tree` \*\*c\_array, int clusters)
- void `generate_xml` (`decomposed_tree` \*tree, int idp, int idp2, int idp3)
- void `make_combinations_for_cluster` (`decomposed_tree` \*cluster, `llist` \*combs)
- void `report_tree_separators` (`decomposed_tree` \*tree)
- void `report_tree_separators2` (`decomposed_tree` \*tree)
- int `get_comb_node_id` (void \*body, void \*bla)
- void `merge_tree_nodes` (`decomposed_tree` \*root)
- void `reid_decomposed_tree_nodes` (`decomposed_tree` \*root, int \*id)
- void `merge_tree_nodes_rec` (`decomposed_tree` \*root)
- void `merge_with_parent` (`decomposed_tree` \*tree)
- void `print_tree_report` (`decomposed_tree` \*tree, double \*maxratio, int \*maxcv, int \*maxsv)

#### 4.14.1 Detailed Description

Provides the implementation for the tree clustering data-structure and the algorithms for building them.

#### 4.14.2 Typedef Documentation

4.14.2.1 typedef struct `decomposed_tree` `decomposed_tree`

4.14.2.2 typedef struct `decomposition_info` `decomposition_info`

4.14.2.3 typedef struct `node_graph` `node_graph`

4.14.2.4 typedef struct `separator_info` `separator_info`

#### 4.14.3 Function Documentation

4.14.3.1 int `add_affiliate_rels` ( `llist` \* `tree_nodes` )

4.14.3.2 void `add_decomposed_tree_child` ( `decomposed_tree` \* `child`, `decomposed_tree` \* `parent` )

4.14.3.3 void `add_dual_fill_edges` ( char \*\* `matrix`, `constraint_graph_node` \* `node`, `heap` \* `h`, `constraint_graph` \* `cg` )

4.14.3.4 void `add_fill_edges` ( `node_graph` \* `ng`, `heap` \* `h`, int `var`, `llist` \* `added_edges` )

4.14.3.5 void `add_fill_edges_to_node_graph` ( `node_graph` \* `ng`, `llist` \* `filledges` )

4.14.3.6 void `add_separator_relations` ( `decomposed_tree` \*\* `c_array`, int `clusters` )

4.14.3.7 void `add_separator_relations_for_cluster` ( `decomposed_tree` \* `current`, int `clusters`, int `clear_combdatas` )

4.14.3.8 void `append_to_comb` ( `htable` \* `ht`, `llist` \* `comb` )

4.14.3.9 void `append_to_comb_head` ( `htable` \* `ht`, `llist` \* `comb` )

- 4.14.3.10 `decomposed_tree*` `build_decomposed_tree_vertices ( llist * cliques, llist * tree_nodes )`
- 4.14.3.11 `void` `check_all_constraints_added ( constraints * cons, FILE * f )`
- 4.14.3.12 `void` `clear_combdatas_from_relations ( decomposed_tree * root )`
- 4.14.3.13 `void` `clear_combdatas_from_relations_only_root ( decomposed_tree * root, int onlyroot )`
- 4.14.3.14 `llist*` `cliques ( int * ordering, node_graph * ng, int * size )`
- 4.14.3.15 `void` `cluster_pre_process ( decomposed_tree * tree, int copy_tables )`
- 4.14.3.16 `void` `cluster_pre_process2 ( decomposed_tree * tree, int copy_tables )`
- 4.14.3.17 `int` `count_dual_fill_edges ( char ** matrix, constraint_graph_node * node )`
- 4.14.3.18 `int` `count_fill_edges ( node_graph * ng, int var )`
- 4.14.3.19 `void` `destroy_node_graph ( node_graph * ng )`
- 4.14.3.20 `llist*` `dual_cliques ( constraint_graph_node ** ordering, constraint_graph * cg )`
- 4.14.3.21 `int` `filter_tuples_for_var_val ( variable * var, value * val )`
- 4.14.3.22 `decomposed_tree*` `find_balancing_root ( llist * nodes )`
- 4.14.3.23 `decomposed_tree*` `find_max_degree_node ( llist * nodes )`
- 4.14.3.24 `decomposed_tree*` `find_smallest_root ( llist * nodes )`
- 4.14.3.25 `void` `generate_xml ( decomposed_tree * tree, int idp, int idp2, int idp3 )`
- 4.14.3.26 `decomposed_tree**` `get_cluster_array ( decomposed_tree * root )`
- 4.14.3.27 `int` `get_comb_node_id ( void * body, void * bla )`
- 4.14.3.28 `int` `get_common_rels_count_from_sets ( set * va, set * vb, llist * clist )`
- 4.14.3.29 `int` `get_common_vars_count_from_sets ( set * va, set * vb, llist * clist )`
- 4.14.3.30 `decomposed_tree*` `get_decomposed_tree_from_library ( constraint_graph * cg, llist * tree_nodes, int tree_only_no_cg )`
- 4.14.3.31 `decomposed_tree*` `get_decomposed_tree_from_library_rec ( tree_node * tree, constraint_graph * cg, llist * tree_nodes, int tree_only_no_cg )`
- 4.14.3.32 `decomposed_tree*` `get_dual_decomposed_tree_from_library ( constraint_graph * cg, llist * tree_nodes, int tree_only_no_cg )`
- 4.14.3.33 `decomposed_tree*` `get_dual_decomposed_tree_from_library_rec ( tree_node * tree, constraint_graph * cg, llist * tree_nodes, int tree_only_no_cg )`
- 4.14.3.34 `constraint_graph_node**` `get_dual_elimination_ordering ( constraint_graph * cg )`
- 4.14.3.35 `int` `get_dual_heap_value ( void * body )`

- 4.14.3.36 `int* get_elimination_ordering ( node_graph * ng, llist * added_edges )`
- 4.14.3.37 `int get_heap_value ( void * body )`
- 4.14.3.38 `void hide_comb_datas ( decomposed_tree * node )`
- 4.14.3.39 `int in_subtree ( decomposed_tree * root, decomposed_tree * node )`
- 4.14.3.40 `void init_cluster_queue ( decomposed_tree * tree )`
- 4.14.3.41 `void label_depth ( decomposed_tree * root )`
- 4.14.3.42 `int label_levels ( decomposed_tree * root )`
- 4.14.3.43 `void make_combinations_for_cluster ( decomposed_tree * cluster, llist * combs )`
- 4.14.3.44 `void make_graph ( decomposed_tree * root, llist * leafs, decomposed_tree *** array )`
- 4.14.3.45 `void make_this_root ( decomposed_tree * newroot )`
- 4.14.3.46 `int* max_cardinality ( node_graph * ng, int node )`
- 4.14.3.47 `void merge_tree_nodes ( decomposed_tree * root )`
- 4.14.3.48 `void merge_tree_nodes_rec ( decomposed_tree * root )`
- 4.14.3.49 `void merge_with_parent ( decomposed_tree * treen )`
- 4.14.3.50 `decomposed_tree* new_decomposed_tree ( llist * vl )`
- 4.14.3.51 `decomposition_info* new_di ( void )`
- 4.14.3.52 `node_graph* new_node_graph ( constraint_graph * cg )`
- 4.14.3.53 `node_graph* new_node_graph_from_rels ( llist * rels, llist * vars, int * varmap )`
- 4.14.3.54 `node_graph* new_node_graph_wpeo ( constraint_graph * cg, node_graph * old_graph_get_node_peo_from, llist * added_edges )`
- 4.14.3.55 `int ng_get_heap_value ( void * body )`
- 4.14.3.56 `void ng_set_heap_pos ( int pos, void * body )`
- 4.14.3.57 `void ng_set_heap_value ( int val, void * body )`
- 4.14.3.58 `void normalized_projected_rels_on_c_vars ( decomposed_tree * treen )`
- 4.14.3.59 `void print_decomposed_tree ( decomposed_tree * tree, int tab, FILE * f, int summary, int print_screen )`
- 4.14.3.60 `void print_decomposed_tree_cluster ( decomposed_tree * tree )`
- 4.14.3.61 `void print_decomposed_tree_cluster_dot ( decomposed_tree * tree )`
- 4.14.3.62 `void print_decomposed_tree_node ( decomposed_tree * tree, int tab, decomposition_info * di )`
- 4.14.3.63 `void print_decomposed_tree_node_dot ( decomposed_tree * tree, decomposition_info * di, FILE * f )`

- 4.14.3.64 void print\_decomposed\_tree\_node\_summary\_dot ( decomposed\_tree \* tree, decomposition\_info \* di, FILE \* f )
- 4.14.3.65 void print\_decomposed\_tree\_rec ( decomposed\_tree \* tree, int tab, int summary, decomposition\_info \* di, FILE \* f )
- 4.14.3.66 void print\_decomposition\_tree\_tables ( decomposed\_tree \* treen )
- 4.14.3.67 void print\_decomposition\_tree\_tables\_h ( decomposed\_tree \* treen )
- 4.14.3.68 void print\_filtering\_report ( constraints \* cons )
- 4.14.3.69 void print\_node\_graph ( node\_graph \* ng )
- 4.14.3.70 void print\_tree\_report ( decomposed\_tree \* tree, double \* maxratio, int \* maxcv, int \* maxsv )
- 4.14.3.71 void reid\_decomposed\_tree\_nodes ( decomposed\_tree \* root, int \* id )
- 4.14.3.72 void report\_separator\_quality ( decomposed\_tree \* child )
- 4.14.3.73 void report\_tree\_separators ( decomposed\_tree \* treen )
- 4.14.3.74 void report\_tree\_separators2 ( decomposed\_tree \* treen )
- 4.14.3.75 void set\_dual\_heap\_position ( int pos, void \* body )
- 4.14.3.76 void set\_dual\_heap\_value ( int val, void \* body )
- 4.14.3.77 void set\_heap\_position ( int pos, void \* body )
- 4.14.3.78 void set\_leafs ( decomposed\_tree \* root )
- 4.14.3.79 void set\_pre\_post ( decomposed\_tree \* treen, int \* id, decomposed\_tree \*\* treens )
- 4.14.3.80 int set\_subtree\_size ( decomposed\_tree \* root )
- 4.14.3.81 void set\_uncovered\_variables ( decomposed\_tree \* root )
- 4.14.3.82 int setup\_combinations ( decomposed\_tree \*\* c\_array, int clusters )
- 4.14.3.83 void split\_trees ( decomposed\_tree \* tree, llist \* trees )
- 4.14.3.84 void swap\_parent\_child ( decomposed\_tree \* child, decomposed\_tree \* parent )
- 4.14.3.85 void un\_hide\_comb\_datas ( decomposed\_tree \* node )

## 4.15 decomposition\_functions.c File Reference

```
#include "decomposition_functions.h"
#include "decomposition.h"
#include "query_graph.h"
#include "utils.h"
#include "combinations_wrapper.h"
#include "rs2.h"
#include "tree_decomp.h"
#include "rsprocedures.h"
#include "tree.h"
#include "active_tuple.h"
#include "splitter.h"
#include "minibucket.h"
#include "partition.h"
#include "hashtable_search.h"
#include <math.h>
#include "rs2tree.h"
#include "tables.h"
```

### Functions

- void [add\\_constraints\\_to\\_cluique\\_pD\\_old\\_not\\_used](#) ([decomposed\\_tree](#) \*clique, [constraint\\_graph](#) \*cg)
- void [add\\_constraints\\_to\\_cluique\\_pE](#) ([decomposed\\_tree](#) \*clique, [constraint\\_graph](#) \*cg, int filter\_from\_omitted)
- void [add\\_constraints\\_to\\_cluique\\_pB](#) ([decomposed\\_tree](#) \*clique)
- void [add\\_constraints\\_to\\_cluique\\_pA](#) ([decomposed\\_tree](#) \*clique, [constraint\\_graph](#) \*cg)
- void [add\\_constraints\\_to\\_cluique\\_pC](#) ([decomposed\\_tree](#) \*clique, [constraint\\_graph](#) \*cg)
- void [add\\_constraints\\_to\\_cluiques](#) ([decomposed\\_tree](#) \*tree, [llist](#) \*added\_edges, [constraint\\_graph](#) \*cg)
- int [process\\_cluster\\_generate\\_separator](#) ([decomposed\\_tree](#) \*treen)
- int [process\\_cluster\\_filter\\_separators](#) ([decomposed\\_tree](#) \*treen)
- void [generate\\_solutions](#) ([decomposed\\_tree](#) \*treen, [active\\_tuple](#) \*ac)
- void [generate\\_solutions2](#) ([decomposed\\_tree](#) \*treen, [active\\_tuple](#) \*ac)
- int [combinations\\_splitter](#) ([decomposed\\_tree](#) \*treen)
- void [set\\_all\\_relation\\_scope\\_separator](#) ([decomposed\\_tree](#) \*treen)
- void [isolate\\_relations](#) ([decomposed\\_tree](#) \*treen)

### 4.15.1 Function Documentation

4.15.1.1 void [add\\_constraints\\_to\\_cluique\\_pA](#) ( [decomposed\\_tree](#) \* clique, [constraint\\_graph](#) \* cg )

4.15.1.2 void [add\\_constraints\\_to\\_cluique\\_pB](#) ( [decomposed\\_tree](#) \* clique )

4.15.1.3 void [add\\_constraints\\_to\\_cluique\\_pC](#) ( [decomposed\\_tree](#) \* clique, [constraint\\_graph](#) \* cg )

4.15.1.4 void [add\\_constraints\\_to\\_cluique\\_pD\\_old\\_not\\_used](#) ( [decomposed\\_tree](#) \* clique, [constraint\\_graph](#) \* cg )

4.15.1.5 void [add\\_constraints\\_to\\_cluique\\_pE](#) ( [decomposed\\_tree](#) \* clique, [constraint\\_graph](#) \* cg, int filter\_from\_omitted )

4.15.1.6 void [add\\_constraints\\_to\\_cluiques](#) ( [decomposed\\_tree](#) \* tree, [llist](#) \* added\_edges, [constraint\\_graph](#) \* cg )

4.15.1.7 int [combinations\\_splitter](#) ( [decomposed\\_tree](#) \* treen )



- 4.15.1.8 void generate\_solutions ( decomposed\_tree \* *treen*, active\_tuple \* *ac* )
- 4.15.1.9 void generate\_solutions2 ( decomposed\_tree \* *treen*, active\_tuple \* *ac* )
- 4.15.1.10 void isolate\_relations ( decomposed\_tree \* *treen* )
- 4.15.1.11 int process\_cluster\_filter\_separators ( decomposed\_tree \* *treen* )
- 4.15.1.12 int process\_cluster\_generate\_separator ( decomposed\_tree \* *treen* )
- 4.15.1.13 void set\_all\_relation\_scope\_separator ( decomposed\_tree \* *treen* )

## 4.16 decomposition\_functions.h File Reference

```
#include "utils.h"
#include "decomposition.h"
```

### Functions

- void [add\\_constraints\\_to\\_cluique\\_pA](#) (decomposed\_tree \**clique*, constraint\_graph \**cg*)
- void [add\\_constraints\\_to\\_cluique\\_pD\\_old\\_not\\_used](#) (decomposed\_tree \**clique*, constraint\_graph \**cg*)
- void [add\\_constraints\\_to\\_cluique\\_pD](#) (decomposed\_tree \**clique*, constraint\_graph \**cg*)
- void [add\\_constraints\\_to\\_cluique\\_pB](#) (decomposed\_tree \**clique*)
- void [add\\_constraints\\_to\\_cluique\\_pC](#) (decomposed\_tree \**clique*, constraint\_graph \**cg*)
- void [add\\_constraints\\_to\\_cluique\\_pE](#) (decomposed\_tree \**clique*, constraint\_graph \**cg*, int filter\_from\_omitted)
- void [add\\_constraints\\_to\\_cluiques](#) (decomposed\_tree \**tree*, llist \**added\_edges*, constraint\_graph \**cg*)
- int [process\\_cluster\\_generate\\_separator](#) (decomposed\_tree \**treen*)
- int [process\\_cluster\\_filter\\_separators](#) (decomposed\_tree \**treen*)
- void [generate\\_solutions](#) (decomposed\_tree \**treen*, active\_tuple \**ac*)
- void [generate\\_solutions2](#) (decomposed\_tree \**treen*, active\_tuple \**ac*)
- int [combinations\\_splitter](#) (decomposed\_tree \**treen*)
- void [set\\_all\\_relation\\_scope\\_separator](#) (decomposed\_tree \**treen*)
- void [isolate\\_relations](#) (decomposed\_tree \**tree*)

### 4.16.1 Detailed Description

Implements helper methods used with the tree clustering.

### 4.16.2 Function Documentation

- 4.16.2.1 void [add\\_constraints\\_to\\_cluique\\_pA](#) ( decomposed\_tree \* *clique*, constraint\_graph \* *cg* )
- 4.16.2.2 void [add\\_constraints\\_to\\_cluique\\_pB](#) ( decomposed\_tree \* *clique* )
- 4.16.2.3 void [add\\_constraints\\_to\\_cluique\\_pC](#) ( decomposed\_tree \* *clique*, constraint\_graph \* *cg* )
- 4.16.2.4 void [add\\_constraints\\_to\\_cluique\\_pD](#) ( decomposed\_tree \* *clique*, constraint\_graph \* *cg* )
- 4.16.2.5 void [add\\_constraints\\_to\\_cluique\\_pD\\_old\\_not\\_used](#) ( decomposed\_tree \* *clique*, constraint\_graph \* *cg* )

4.16.2.6 void add\_constraints\_to\_clique\_pE ( decomposed\_tree \* clique, constraint\_graph \* cg, int filter\_from\_omitted )

4.16.2.7 void add\_constraints\_to\_cliques ( decomposed\_tree \* tree, llist \* added\_edges, constraint\_graph \* cg )

4.16.2.8 int combinations\_splitter ( decomposed\_tree \* treen )

4.16.2.9 void generate\_solutions ( decomposed\_tree \* treen, active\_tuple \* ac )

4.16.2.10 void generate\_solutions2 ( decomposed\_tree \* treen, active\_tuple \* ac )

4.16.2.11 void isolate\_relations ( decomposed\_tree \* tree )

4.16.2.12 int process\_cluster\_filter\_separators ( decomposed\_tree \* treen )

4.16.2.13 int process\_cluster\_generate\_separator ( decomposed\_tree \* treen )

4.16.2.14 void set\_all\_relation\_scope\_separator ( decomposed\_tree \* treen )

## 4.17 do\_db.c File Reference

```
#include "do_db.h"
#include "utils.h"
#include "query_graph.h"
#include "rpwc.h"
#include "rs.h"
#include "rsprocedures.h"
#include "rpwcprocedures.h"
#include "combinations_wrapper.h"
#include "rs2.h"
#include "tree.h"
#include "tree_decomp.h"
#include "db_algo1.h"
#include "decomposition.h"
#include "btdallsol.h"
#include "splitter.h"
#include "decomposition_functions.h"
#include "rs2tree.h"
#include "septables.h"
#include "gbtd.h"
```

### Functions

- int [do\\_db](#) (main\_structure \*m\_s)
- int [do\\_tree](#) (main\_structure \*m\_s)

#### 4.17.1 Function Documentation

4.17.1.1 int do\_db ( main\_structure \* m\_s )

4.17.1.2 int do\_tree ( main\_structure \* m\_s )

## 4.18 do\_db.h File Reference

```
#include "globals.h"
```

### Functions

- int [do\\_db](#) ([main\\_structure](#) \*m\_s)
- int [do\\_tree](#) ([main\\_structure](#) \*m\_s)

### 4.18.1 Detailed Description

Provides the interfaces for initializing the tree-decomposition.

### 4.18.2 Function Documentation

4.18.2.1 int [do\\_db](#) ( [main\\_structure](#) \* *m\_s* )

4.18.2.2 int [do\\_tree](#) ( [main\\_structure](#) \* *m\_s* )

## 4.19 domains.c File Reference

```
#include "domains.h"  
#include "utils.h"
```

### Functions

- [llist](#) \* [new\\_domains](#) (void)
- void [add\\_domain\\_name](#) ([llist](#) \*domains, char \*name)
- void [add\\_domain\\_int\\_list\\_to\\_last\\_domain](#) ([llist](#) \*domains, int from, int to)
- [value](#) \* [value\\_copy](#) ([value](#) \*val)

### 4.19.1 Function Documentation

4.19.1.1 void [add\\_domain\\_int\\_list\\_to\\_last\\_domain](#) ( [llist](#) \* *domains*, int *from*, int *to* )

4.19.1.2 void [add\\_domain\\_name](#) ( [llist](#) \* *domains*, char \* *name* )

4.19.1.3 [llist](#)\* [new\\_domains](#) ( void )

4.19.1.4 [value](#)\* [value\\_copy](#) ( [value](#) \* *val* )

## 4.20 domains.h File Reference

```
#include "llist.h"
```

### Classes

- struct [value](#)

## Typedefs

- typedef struct [value](#) [value](#)

## Functions

- [llist](#) \* [new\\_domains](#) (void)
- void [add\\_domain\\_name](#) ([llist](#) \*domains, char \*name)
- void [add\\_domain\\_int\\_list\\_to\\_last\\_domain](#) ([llist](#) \*domains, int from, int to)
- [value](#) \* [value\\_copy](#) ([value](#) \*val)

### 4.20.1 Detailed Description

Provides the data structures for representing and domains of variables.

### 4.20.2 Typedef Documentation

#### 4.20.2.1 typedef struct [value](#) [value](#)

### 4.20.3 Function Documentation

#### 4.20.3.1 void [add\\_domain\\_int\\_list\\_to\\_last\\_domain](#) ( [llist](#) \* *domains*, int *from*, int *to* )

#### 4.20.3.2 void [add\\_domain\\_name](#) ( [llist](#) \* *domains*, char \* *name* )

#### 4.20.3.3 [llist](#)\* [new\\_domains](#) ( void )

#### 4.20.3.4 [value](#)\* [value\\_copy](#) ( [value](#) \* *val* )

## 4.21 gbtd.c File Reference

```
#include "gbtd.h"
#include "utils.h"
#include "variables.h"
#include "rs2tree.h"
#include "rs2.h"
#include "rpscprocedures.h"
#include "btdallsol.h"
```

## Functions

- [gbtd\\_system](#) \* [new\\_gbtd\\_system](#) (void)
- [pseudo\\_cluster](#) \* [new\\_pseudo\\_cluster](#) (void)
- void [destroy\\_pseudo\\_cluster](#) ([pseudo\\_cluster](#) \*ps)
- [gbtd\\_system](#) \* [init\\_gbtd](#) ([llist](#) \*vars, int allSol, [options](#) system, [set](#) \*queue, [set](#) \*\*queues, int total\_clusters, [decomposed\\_tree](#) \*\*treens, [comb\\_data](#) \*cd, int revision\_time, int \*tuple\_positions, int \*tuple\_position\_htid, [active\\_tuple](#) \*ac, int count\_on\_m\_s)
- void [destroy\\_gbtd\\_system](#) ([gbtd\\_system](#) \*gs)
- void [generate\\_pseudo\\_tree](#) ([gbtd\\_system](#) \*gs)
- int [run\\_gbtd](#) ([gbtd\\_system](#) \*gs)
- int [run\\_gbtd\\_wtn](#) ([gbtd\\_system](#) \*gs)
- int [current\\_has\\_next](#) ([gbtd\\_system](#) \*gs)

- int `filter_gvars` (`gbtd_system *gs`, int `skip_if_all_revised`)
- `pseudo_cluster *sibling` (`pseudo_cluster *pc`)
- void `find_gvar` (`gbtd_system *gs`)
- void `find_gvar_wtn` (`gbtd_system *gs`, int `backwards`)
- void `advance_current_domain` (`gbtd_system *gs`)
- void `destroy_var_undo_data` (`var_undo_data *vud`)
- void `move_to_previous_var` (`gbtd_system *gs`)
- void `reset_gvar` (`gbtd_system *gs`, void `*gvar`)
- void `assign_gvar` (`gbtd_system *gs`)
- `htv_undo_data *new_htv_undo_data` (`htvar *htv`, `gbtd_system *gs`)
- `var_undo_data *new_var_undo_data` (`variable *var`)
- int `unassign_subtree` (`gbtd_system *gs`)
- void `print_gbtd_system` (`gbtd_system *gs`)
- void `undo_htv_assignment` (`htv_undo_data *vud`, `gbtd_system *gs`)
- `htvar *new_htvar` (`htable *ht`, int `id`, `llist_node *chosen_value`)
- void `destroy_htvar` (`htvar *htv`)
- void `do_something_with_solution` (`gbtd_system *gs`)
- void `init_goods_nogoods` (`pseudo_cluster *psc`)
- void `set_good_nogood` (`pseudo_cluster *psc`)
- int `has_nogood` (`pseudo_cluster *psc`)
- double `has_good` (`pseudo_cluster *psc`)
- int `relation_var_tuple_stamp` (`gbtd_system *gs`, `htvar *htv`, int `*tuple`, int `first`, int `marked`)
- int `covers_sep_relations` (`htvar *htv`, `gbtd_system *gs`)
- int `no_unassigned_cover_variable` (`gbtd_system *gs`)
- void `count_goods_nogoods` (`gbtd_system *gs`)
- void `count_goods_nogoods_rec` (`pseudo_cluster *root`, double `*countnogood`, double `*countgood`)

## 4.21.1 Function Documentation

4.21.1.1 void `advance_current_domain` ( `gbtd_system * gs` )

4.21.1.2 void `assign_gvar` ( `gbtd_system * gs` )

4.21.1.3 void `count_goods_nogoods` ( `gbtd_system * gs` )

4.21.1.4 void `count_goods_nogoods_rec` ( `pseudo_cluster * root`, double `* countnogood`, double `* countgood` )

4.21.1.5 int `covers_sep_relations` ( `htvar * htv`, `gbtd_system * gs` )

4.21.1.6 int `current_has_next` ( `gbtd_system * gs` )

4.21.1.7 void `destroy_gbtd_system` ( `gbtd_system * gs` )

4.21.1.8 void `destroy_htvar` ( `htvar * htv` )

4.21.1.9 void `destroy_pseudo_cluster` ( `pseudo_cluster * ps` )

4.21.1.10 void `destroy_var_undo_data` ( `var_undo_data * vud` )

4.21.1.11 void `do_something_with_solution` ( `gbtd_system * gs` )

4.21.1.12 int `filter_gvars` ( `gbtd_system * gs`, int `skip_if_all_revised` )

4.21.1.13 void `find_gvar` ( `gbtd_system * gs` )

- 4.21.1.14 void find\_gvar\_wtn ( gbtd\_system \* gs, int backwards )
- 4.21.1.15 void generate\_pseudo\_tree ( gbtd\_system \* gs )
- 4.21.1.16 double has\_good ( pseudo\_cluster \* psc )
- 4.21.1.17 int has\_nogood ( pseudo\_cluster \* psc )
- 4.21.1.18 gbtd\_system\* init\_gbtd ( llist \* vars, int allSol, options system, set \* queue, set \*\* queues, int total\_clusters, decomposed\_tree \*\* treens, comb\_data \* cd, int revision\_time, int \* tuple\_positions, int \* tuple\_position\_htid, active\_tuple \* ac, int count\_on\_m\_s )
- 4.21.1.19 void init\_goods\_nogoods ( pseudo\_cluster \* psc )
- 4.21.1.20 void move\_to\_previous\_var ( gbtd\_system \* gs )
- 4.21.1.21 gbtd\_system\* new\_gbtd\_system ( void )
- 4.21.1.22 htv\_undo\_data\* new\_htv\_undo\_data ( htv \* htv, gbtd\_system \* gs )
- 4.21.1.23 htv\* new\_htv ( htable \* ht, int id, llist\_node \* chosen\_value )
- 4.21.1.24 pseudo\_cluster\* new\_pseudo\_cluster ( void )
- 4.21.1.25 var\_undo\_data\* new\_var\_undo\_data ( variable \* var )
- 4.21.1.26 int no\_unassigned\_cover\_variable ( gbtd\_system \* gs )
- 4.21.1.27 void print\_gbtd\_system ( gbtd\_system \* gs )
- 4.21.1.28 int relation\_var\_tuple\_stamp ( gbtd\_system \* gs, htv \* htv, int \* tuple, int first, int marked )
- 4.21.1.29 void reset\_gvar ( gbtd\_system \* gs, void \* gvar )
- 4.21.1.30 int run\_gbtd ( gbtd\_system \* gs )
- 4.21.1.31 int run\_gbtd\_wtn ( gbtd\_system \* gs )
- 4.21.1.32 void set\_good\_nogood ( pseudo\_cluster \* psc )
- 4.21.1.33 pseudo\_cluster\* sibling ( pseudo\_cluster \* pc )
- 4.21.1.34 int unassign\_subtree ( gbtd\_system \* gs )
- 4.21.1.35 void undo\_htv\_assignment ( htv\_undo\_data \* vud, gbtd\_system \* gs )

## 4.22 gbtd.h File Reference

```
#include "rb_tree.h"
#include "llist.h"
#include "decomposition.h"
```

## Classes

- struct [pseudo\\_cluster](#)
- struct [gbtd\\_system](#)
- struct [var\\_undo\\_data](#)
- struct [htvar](#)
- struct [htv\\_undo\\_data](#)

## Typedefs

- typedef struct [pseudo\\_cluster](#) [pseudo\\_cluster](#)
- typedef struct [gbtd\\_system](#) [gbtd\\_system](#)
- typedef struct [var\\_undo\\_data](#) [var\\_undo\\_data](#)
- typedef struct [htvar](#) [htvar](#)
- typedef struct [htv\\_undo\\_data](#) [htv\\_undo\\_data](#)

## Functions

- [gbtd\\_system](#) \* [init\\_gbtd](#) ([llist](#) \*vars, int allSol, [options](#) system, [set](#) \*queue, [set](#) \*\*queues, int total\_clusters, [decomposed\\_tree](#) \*\*treens, [comb\\_data](#) \*cd, int revision\_time, int \*tuple\_positions, int \*tuple\_position\_htid, [active\\_tuple](#) \*ac, int count\_on\_m\_s)
- void [move\\_to\\_previous\\_var](#) ([gbtd\\_system](#) \*gs)
- [pseudo\\_cluster](#) \* [sibling](#) ([pseudo\\_cluster](#) \*pc)
- int [unassign\\_subtree](#) ([gbtd\\_system](#) \*gs)
- void [assign\\_gvar](#) ([gbtd\\_system](#) \*gs)
- int [filter\\_gvars](#) ([gbtd\\_system](#) \*gs, int skip\_if\_all\_revised)
- void [advance\\_current\\_domain](#) ([gbtd\\_system](#) \*gs)
- void [reset\\_gvar](#) ([gbtd\\_system](#) \*gs, void \*gvar)
- void [find\\_gvar](#) ([gbtd\\_system](#) \*gs)
- void [find\\_gvar\\_wtn](#) ([gbtd\\_system](#) \*gs, int backwards)
- [gbtd\\_system](#) \* [new\\_gbtd\\_system](#) (void)
- [pseudo\\_cluster](#) \* [new\\_pseudo\\_cluster](#) (void)
- void [generate\\_pseudo\\_tree](#) ([gbtd\\_system](#) \*gs)
- int [run\\_gbtd](#) ([gbtd\\_system](#) \*gs)
- int [run\\_gbtd\\_wtn](#) ([gbtd\\_system](#) \*gs)
- int [current\\_has\\_next](#) ([gbtd\\_system](#) \*gs)
- void [destroy\\_var\\_undo\\_data](#) ([var\\_undo\\_data](#) \*vud)
- [var\\_undo\\_data](#) \* [new\\_var\\_undo\\_data](#) ([variable](#) \*var)
- void [print\\_gbtd\\_system](#) ([gbtd\\_system](#) \*gs)
- void [destroy\\_pseudo\\_cluster](#) ([pseudo\\_cluster](#) \*ps)
- void [destroy\\_gbtd\\_system](#) ([gbtd\\_system](#) \*gs)
- void [undo\\_htv\\_assignment](#) ([htv\\_undo\\_data](#) \*vud, [gbtd\\_system](#) \*gs)
- [htv\\_undo\\_data](#) \* [new\\_htv\\_undo\\_data](#) ([htvar](#) \*htv, [gbtd\\_system](#) \*gs)
- [htvar](#) \* [new\\_htvar](#) ([htable](#) \*ht, int id, [llist\\_node](#) \*chosen\_value)
- void [destroy\\_htvar](#) ([htvar](#) \*htv)
- void [do\\_something\\_with\\_solution](#) ([gbtd\\_system](#) \*gs)
- double [has\\_good](#) ([pseudo\\_cluster](#) \*psc)
- int [has\\_nogood](#) ([pseudo\\_cluster](#) \*psc)
- void [set\\_good\\_nogood](#) ([pseudo\\_cluster](#) \*psc)
- void [init\\_goods\\_nogoods](#) ([pseudo\\_cluster](#) \*psc)
- int [relation\\_var\\_tuple\\_stamp](#) ([gbtd\\_system](#) \*gs, [htvar](#) \*htv, int \*tuple, int first, int marked)
- int [no\\_unassigned\\_cover\\_variable](#) ([gbtd\\_system](#) \*gs)
- int [covers\\_sep\\_relations](#) ([htvar](#) \*htv, [gbtd\\_system](#) \*gs)
- void [count\\_goods\\_nogoods\\_rec](#) ([pseudo\\_cluster](#) \*root, double \*countnogood, double \*countgood)
- void [count\\_goods\\_nogoods](#) ([gbtd\\_system](#) \*gs)

### 4.22.1 Detailed Description

Provides a generalized backtracking with tree decomposition algorithm. Used for most backtrack-based algorithms with and without tree-decomposition both for variables and relations. Also implements the core of PerTuple and AllSol algorithms.

### 4.22.2 Typedef Documentation

4.22.2.1 typedef struct gbtd\_system gbtd\_system

4.22.2.2 typedef struct htv\_undo\_data htv\_undo\_data

4.22.2.3 typedef struct htvar htvar

4.22.2.4 typedef struct pseudo\_cluster pseudo\_cluster

4.22.2.5 typedef struct var\_undo\_data var\_undo\_data

### 4.22.3 Function Documentation

4.22.3.1 void advance\_current\_domain ( gbtd\_system \* gs )

4.22.3.2 void assign\_gvar ( gbtd\_system \* gs )

4.22.3.3 void count\_goods\_nogoods ( gbtd\_system \* gs )

4.22.3.4 void count\_goods\_nogoods\_rec ( pseudo\_cluster \* root, double \* countnogood, double \* countgood )

4.22.3.5 int covers\_sep\_relations ( htvar \* htv, gbtd\_system \* gs )

4.22.3.6 int current\_has\_next ( gbtd\_system \* gs )

4.22.3.7 void destroy\_gbtd\_system ( gbtd\_system \* gs )

4.22.3.8 void destroy\_htvar ( htvar \* htv )

4.22.3.9 void destroy\_pseudo\_cluster ( pseudo\_cluster \* ps )

4.22.3.10 void destroy\_var\_undo\_data ( var\_undo\_data \* vud )

4.22.3.11 void do\_something\_with\_solution ( gbtd\_system \* gs )

4.22.3.12 int filter\_gvars ( gbtd\_system \* gs, int skip\_if\_all\_revised )

4.22.3.13 void find\_gvar ( gbtd\_system \* gs )

4.22.3.14 void find\_gvar\_wtn ( gbtd\_system \* gs, int backwards )

4.22.3.15 void generate\_pseudo\_tree ( gbtd\_system \* gs )

4.22.3.16 double has\_good ( pseudo\_cluster \* psc )

4.22.3.17 int has\_nogood ( pseudo\_cluster \* psc )



- 4.22.3.18 `gbtd_system*` `init_gbtd ( llist * vars, int allSol, options system, set * queue, set ** queues, int total_clusters, decomposed_tree ** treens, comb_data * cd, int revision_time, int * tuple_positions, int * tuple_position_htid, active_tuple * ac, int count_on_m_s )`
- 4.22.3.19 `void` `init_goods_nogoods ( pseudo_cluster * psc )`
- 4.22.3.20 `void` `move_to_previous_var ( gbtd_system * gs )`
- 4.22.3.21 `gbtd_system*` `new_gbtd_system ( void )`
- 4.22.3.22 `htv_undo_data*` `new_htv_undo_data ( htvar * htv, gbtd_system * gs )`
- 4.22.3.23 `htvar*` `new_htvar ( htable * ht, int id, llist_node * chosen_value )`
- 4.22.3.24 `pseudo_cluster*` `new_pseudo_cluster ( void )`
- 4.22.3.25 `var_undo_data*` `new_var_undo_data ( variable * var )`
- 4.22.3.26 `int` `no_unassigned_cover_variable ( gbtd_system * gs )`
- 4.22.3.27 `void` `print_gbtd_system ( gbtd_system * gs )`
- 4.22.3.28 `int` `relation_var_tuple_stamp ( gbtd_system * gs, htvar * htv, int * tuple, int first, int marked )`
- 4.22.3.29 `void` `reset_gvar ( gbtd_system * gs, void * gvar )`
- 4.22.3.30 `int` `run_gbtd ( gbtd_system * gs )`
- 4.22.3.31 `int` `run_gbtd_wtn ( gbtd_system * gs )`
- 4.22.3.32 `void` `set_good_nogood ( pseudo_cluster * psc )`
- 4.22.3.33 `pseudo_cluster*` `sibling ( pseudo_cluster * pc )`
- 4.22.3.34 `int` `unassign_subtree ( gbtd_system * gs )`
- 4.22.3.35 `void` `undo_htv_assignment ( htv_undo_data * vud, gbtd_system * gs )`

## 4.23 globals.c File Reference

```
#include "globals.h"
#include "utils.h"
#include "solution.h"
```

### Functions

- `int` `get_partition_id ()`
- `void` `into_components_problem (main_structure *m_s)`
- `void` `set_m_s_basedon_this (main_structure *newms, main_structure *m_s)`
- `void` `add_single_variable_component (int varnumber, main_structure *m_s)`
- `void` `add_component (int vars, int rels, main_structure *m_s, llist *partition)`

### 4.23.1 Function Documentation

4.23.1.1 void `add_component` ( int *vars*, int *rels*, `main_structure * m_s`, `llist * partition` )

4.23.1.2 void `add_single_variable_component` ( int *varnumber*, `main_structure * m_s` )

4.23.1.3 int `get_partition_id` ( void )

4.23.1.4 void `into_components_problem` ( `main_structure * m_s` )

4.23.1.5 void `set_m_s_basedon_this` ( `main_structure * newms`, `main_structure * m_s` )

## 4.24 globals.h File Reference

```
#include <stdlib.h>
#include "variables.h"
#include "constraints.h"
#include "relations.h"
#include "variable_tracking.h"
#include "set.h"
#include <time.h>
```

### Classes

- struct [main\\_structure](#)

### Typedefs

- typedef enum [options](#) [options](#)
- typedef struct [main\\_structure](#) [main\\_structure](#)

### Enumerations

- enum [options](#) {  
[EMPTY](#), [PAIR\\_INDEX\\_FC](#), [PAIR\\_INDEX\\_BC](#), [PAIR\\_INDEX\\_ALL\\_SOL](#),  
[PAIR\\_INDEX\\_AUTO](#), [MULTI\\_TUPLE\\_INDEX](#), [SKIP\\_INSOL\\_TUPLES](#), [OLD\\_STUFF](#),  
[MIN\\_FILL](#), [LIB](#), [LIB\\_DUAL](#), [MIN\\_FILL\\_DUAL](#),  
[RS\\_GAC](#), [MIN\\_FILL\\_LIB](#), [NOSPLITSEP](#), [GLOBALSEP](#),  
[COMBSEP](#), [MBESEP](#), [MAXCLIQUESEP](#), [MAXCLIQUESEP2](#),  
[MAXCLIQUESEP\\_PP](#), [BINSEP](#), [BINSEP\\_PP](#), [ALLSCOPESSP](#),  
[SPLTSEP](#), [ALLSOLSEP](#), [ONESOLSEP](#), [RMC](#),  
[RMCSEP](#), [RMCSEP\\_SPLT](#), [RMCSEP\\_SPLT\\_ALL\\_SOL](#), [RSTAR](#),  
[BTD](#), [MBE](#), [MBERs](#), [CDATA\\_TREE\\_SEARCH](#),  
[CDATA\\_JOIN](#), [CDATA\\_IGNORE](#), [CDATA\\_RS\\_COMB](#), [RS2TREE\\_SINGLE\\_GRAPH](#),  
[RS2TREE\\_PER\\_CLUSTER](#), [RS2TREE\\_PER\\_CLUSTER\\_SINGLEQ](#), [RS2TREE\\_PER\\_CLUSTER\\_2](#), [RS2-](#)  
[TREE\\_PER\\_CLUSTER\\_ALL\\_1SOL](#),  
[RS2TREE\\_SINGLE\\_DISTRIBUTED\\_GRAPH](#), [RS2TREE\\_PER\\_PARENT\\_CHILD](#), [RS2TREE\\_TUPLE\\_SE-](#)  
[ARCH](#), [SCOPE\\_SUBSET\\_FIT](#),  
[SCOPE\\_SUBSET\\_FIT\\_NOT\\_IN\\_DESC](#), [SCOPE\\_PROJECT\\_FIT](#), [SCOPE\\_PROJECT\\_BIGGEST\\_FIT](#), [ON-](#)  
[E](#),  
[ALL](#), [AUTO](#), [DEFAULT\\_QUEUE](#), [MANAGED\\_QUEUE\\_1](#),  
[MANAGED\\_QUEUE\\_12](#), [MANAGED\\_QUEUE\\_122](#), [MANAGED\\_QUEUE\\_13](#), [MANAGED\\_QUEUE\\_14](#),  
[MANAGED\\_QUEUE\\_2](#), [MANAGED\\_QUEUE\\_d](#), [MANAGED\\_QUEUE\\_r](#), [MANAGED\\_QUEUE\\_SINGLEQ](#),  
[GBTD\\_SYSTEM\\_VARS\\_SINGLE\\_CLUSTER](#), [GBTD\\_SYSTEM\\_VARS\\_TD](#), [GBTD\\_SYSTEM\\_RELS\\_SING-](#)

```
LE_CLUSTER, USE_BTD,
USE_BTD_WTN, BINCLQ, PROJ, COLLECT_PRINT }
```

## Functions

- int `get_partition_id` (void)
- void `check_ok` (main\_structure \*m\_s)
- void `check_ok2` (variables \*vars)
- void `into_components_problem` (main\_structure \*m\_s)
- void `add_component` (int vars, int rels, main\_structure \*m\_s, llist \*partition)
- void `set_m_s_basedon_this` (main\_structure \*newms, main\_structure \*m\_s)
- void `add_single_variable_component` (int varnumber, main\_structure \*m\_s)

### 4.24.1 Detailed Description

Represents the global parameters controlling the program.

### 4.24.2 Typedef Documentation

4.24.2.1 typedef struct main\_structure main\_structure

4.24.2.2 typedef enum options options

### 4.24.3 Enumeration Type Documentation

4.24.3.1 enum options

Enumerator

```
EMPTY
PAIR_INDEX_FC
PAIR_INDEX_BC
PAIR_INDEX_ALL_SOL
PAIR_INDEX_AUTO
MULTI_TUPLE_INDEX
SKIP_INSOL_TUPLES
OLD_STUFF
MIN_FILL
LIB
LIB_DUAL
MIN_FILL_DUAL
RS_GAC
MIN_FILL_LIB
NOSPLITSEP
GLOBALSEP
COMBSEP
MBESEP
MAXCLIQUESEP
MAXCLIQUESEP2
```

**MAXCLIQUESEP\_PP**  
**BINSEP**  
**BINSEP\_PP**  
**ALLSCOPESSP**  
**SPLTSEP**  
**ALLSOLSEP**  
**ONESOLSEP**  
**RMC**  
**RMCSEP**  
**RMCSEP\_SPLT**  
**RMCSEP\_SPLT\_ALL\_SOL**  
**RSTAR**  
**BTB**  
**MBE**  
**MBEs**  
**CDATA\_TREE\_SEARCH**  
**CDATA\_JOIN**  
**CDATA\_IGNORE**  
**CDATA\_RS\_COMB**  
**RS2TREE\_SINGLE\_GRAPH**  
**RS2TREE\_PER\_CLUSTER**  
**RS2TREE\_PER\_CLUSTER\_SINGLEQ**  
**RS2TREE\_PER\_CLUSTER\_2**  
**RS2TREE\_PER\_CLUSTER\_ALL\_1SOL**  
**RS2TREE\_SINGLE\_DISTRIBUTED\_GRAPH**  
**RS2TREE\_PER\_PARENT\_CHILD**  
**RS2TREE\_TUPLE\_SEARCH**  
**SCOPE\_SUBSET\_FIT**  
**SCOPE\_SUBSET\_FIT\_NOT\_IN\_DESC**  
**SCOPE\_PROJECT\_FIT**  
**SCOPE\_PROJECT\_BIGGEST\_FIT**  
**ONE**  
**ALL**  
**AUTO**  
**DEFAULT\_QUEUE**  
**MANAGED\_QUEUE\_1**  
**MANAGED\_QUEUE\_12**  
**MANAGED\_QUEUE\_122**  
**MANAGED\_QUEUE\_13**  
**MANAGED\_QUEUE\_14**  
**MANAGED\_QUEUE\_2**  
**MANAGED\_QUEUE\_d**  
**MANAGED\_QUEUE\_r**  
**MANAGED\_QUEUE\_SINGLEQ**  
**GBTD\_SYSTEM\_VARS\_SINGLE\_CLUSTER**

**GBTD\_SYSTEM\_VARS\_TD**  
**GBTD\_SYSTEM\_RELS\_SINGLE\_CLUSTER**  
**USE\_BT**  
**USE\_BT\_WTN**  
**BINCLQ**  
**PROJ**  
**COLLECT\_PRINT**

#### 4.24.4 Function Documentation

4.24.4.1 void `add_component` ( int *vars*, int *rels*, main\_structure \* *m\_s*, llist \* *partition* )

4.24.4.2 void `add_single_variable_component` ( int *varnumber*, main\_structure \* *m\_s* )

4.24.4.3 void `check_ok` ( main\_structure \* *m\_s* )

4.24.4.4 void `check_ok2` ( variables \* *vars* )

4.24.4.5 int `get_partition_id` ( void )

4.24.4.6 void `into_components_problem` ( main\_structure \* *m\_s* )

4.24.4.7 void `set_m_s_basedon_this` ( main\_structure \* *newms*, main\_structure \* *m\_s* )

## 4.25 hashmap.c File Reference

```

#include "hashmap.h"
#include "utils.h"
#include <math.h>
#include <stdio.h>

```

### Functions

- `hashmap * new_hashmap` (int *size*, int *width*, int \**col\_pos*, int \**max*, int *single*)
- void `destroy_hm` (hashmap \**hm*)
- int `HashDivide` (KEYTYPE *k*, int *m*)
- int `HashMultiply` (KEYTYPE *k*, int *m*)
- int `DoubleProbing` (KEYTYPE *k*, int *i*, int *size*)
- int `tupple_match` (int \**ta*, int \**tb*, hashmap \**hm*, int \**col\_pos*)
- void `insert` (int \**tupple*, hashmap \**hm*)
- llist \* `get` (int \**tupple*, hashmap \**hm*, int \**col\_pos*)
- KEYTYPE `get_key` (int \**tupple*, hashmap \**hm*, int \**col\_pos*)
- void `hm_get_iterator` (hashmap \**hm*, int \**tupple*, int \**col\_pos*)
- int \* `hm_get_next` (hashmap \**hm*)

#### 4.25.1 Function Documentation

4.25.1.1 void `destroy_hm` ( hashmap \* *hm* )

4.25.1.2 int `DoubleProbing` ( KEYTYPE *k*, int *i*, int *size* )

- 4.25.1.3 `llist* get ( int * tuple, hashmap * hm, int * col_pos )`
- 4.25.1.4 `KEYTYPE get_key ( int * tuple, hashmap * hm, int * col_pos )`
- 4.25.1.5 `int HashDivide ( KEYTYPE k, int m )`
- 4.25.1.6 `int HashMultiply ( KEYTYPE k, int m )`
- 4.25.1.7 `void hm_get_iterator ( hashmap * hm, int * tuple, int * col_pos )`
- 4.25.1.8 `int* hm_get_next ( hashmap * hm )`
- 4.25.1.9 `void insert ( int * tuple, hashmap * hm )`
- 4.25.1.10 `hashmap* new_hashmap ( int size, int width, int * col_pos, int * max, int single )`
- 4.25.1.11 `int tuple_match ( int * ta, int * tb, hashmap * hm, int * col_pos )`

## 4.26 hashmap.h File Reference

```
#include "l1ist.h"
```

### Classes

- struct [hashmap](#)

### Macros

- #define [KEYTYPE](#) long long

### Typedefs

- typedef struct [hashmap](#) [hashmap](#)

### Functions

- [hashmap \\* new\\_hashmap](#) (int size, int width, int \*col\_pos, int \*max, int single)
- void [destroy\\_hm](#) (hashmap \*hm)
- int [HashDivide](#) (KEYTYPE k, int m)
- int [HashMultiply](#) (KEYTYPE k, int m)
- int [DoubleProbing](#) (KEYTYPE k, int i, int size)
- void [insert](#) (int \*tuple, hashmap \*hm)
- `l1ist * get` (int \*tuple, hashmap \*hm, int \*col\_pos)
- `KEYTYPE get_key` (int \*tuple, hashmap \*hm, int \*col\_pos)
- int [tuple\\_match](#) (int \*ta, int \*tb, hashmap \*hm, int \*col\_pos)
- void [get\\_iterator](#) (hashmap \*hm, int \*tuple, int \*col\_pos)
- int \* [hm\\_get\\_next](#) (hashmap \*hm)
- void [hm\\_get\\_iterator](#) (hashmap \*hm, int \*tuple, int \*col\_pos)

#### 4.26.1 Detailed Description

Provides a hashmap implementation.

## 4.26.2 Macro Definition Documentation

4.26.2.1 `#define KEYTYPE long long`

## 4.26.3 Typedef Documentation

4.26.3.1 `typedef struct hashmap hashmap`

## 4.26.4 Function Documentation

4.26.4.1 `void destroy_hm ( hashmap * hm )`

4.26.4.2 `int DoubleProbing ( KEYTYPE k, int i, int size )`

4.26.4.3 `llist* get ( int * tuple, hashmap * hm, int * col_pos )`

4.26.4.4 `void get_iterator ( hashmap * hm, int * tuple, int * col_pos )`

4.26.4.5 `KEYTYPE get_key ( int * tuple, hashmap * hm, int * col_pos )`

4.26.4.6 `int HashDivide ( KEYTYPE k, int m )`

4.26.4.7 `int HashMultiply ( KEYTYPE k, int m )`

4.26.4.8 `void hm_get_iterator ( hashmap * hm, int * tuple, int * col_pos )`

4.26.4.9 `int* hm_get_next ( hashmap * hm )`

4.26.4.10 `void insert ( int * tuple, hashmap * hm )`

4.26.4.11 `hashmap* new_hashmap ( int size, int width, int * col_pos, int * max, int single )`

4.26.4.12 `int tuple_match ( int * ta, int * tb, hashmap * hm, int * col_pos )`

## 4.27 hashtable.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "hashtable.h"
#include "hashmap.h"
#include "sort.h"
#include "utils.h"
#include "globals.h"
#include "solution.h"
#include "tree_map.h"
#include "rb_tree.h"
```

### Functions

- `int * get_int_pair ()`
- `htable * new_htable (int id, const char *name, int width, int height, htable *prev, char **col_names, int join_-, cutoff, int is_first, int *vars)`

- `htable * new_htable_replace` (int id, const char \*name, int width, int height, `htable *prev`, char \*\*col\_names, int join\_cutoff, int is\_first, int \*vars, int replace)
- void `add_tuple_list` (`htable *ht`)
- `llist * get_ht_iterator` (`htable *ht`)
- void `clear_deleted_rows` (`htable *ht`)
- char \*\* `copy_col_names` (char \*\*col\_names, int size)
- void `destroy_histogram` (`histogram *histo`)
- `histogram * new_histogram` (int columns)
- int `compare_col_name` (void \*value, void \*bodyb)
- int `compare_col_card` (void \*value, void \*bodyb)
- int `estimate_join_size` (`htable *hta`, `htable *htb`)
- int `estimate_join_size_noproject` (`htable *hta`, `htable *htb`)
- double `ceil` (double x)
- int `estimate_project_size` (`htable *hta`, `llist *project_cols`)
- int `estimate_join_project_size` (`htable *hta`, `htable *htb`, int project)
- void `build_histogram` (`htable *table`)
- void `rebuild_histogram` (`htable *table`)
- void `update_histogram` (`htable *table`)
- `htable * new_clustered_htable` (int id, char \*name, int width, int height, char \*\*col\_names, int join\_cutoff, int \*vars)
- int \* `get_tuple` (int row, `htable *ht`)
- double `get_tuple_value` (int row, `htable *ht`)
- void `set_tuple_value` (double value, int row, `htable *ht`)
- int `get_value` (int row, int column, `htable *ht`)
- int `get_value_sort` (void \*ht, int row, int column)
- void `set_value_sort` (void \*vht, int row, int column, int value)
- void `destroy_htable` (`htable *ht`)
- void `destroy_htable_o` (`htable *ht`, int kill\_col\_names)
- char \*\* `new_names` (int size)
- void `add_name` (char \*name, char \*\*names, int pos)
- `htable * add_tuple` (int \*tuple, `htable *ht`)
- void `print_ht_mysql` (`htable *ht`)
- void `print_ht_h` (`htable *ht`, int short\_name)
- void `print_ht_line` (`htable *ht`)
- void `print_ht` (`htable *ht`, int short\_name)
- void `fprint_ht` (FILE \*file, `htable *ht`, int short\_name)
- void `replace_data` (`htable *data`, `htable *ht`)
- int `have_value` (int x)
- `htable * cluster` (int col1, int col2, `htable *ht`, int no\_destroy)
- `htable * cluster_nosort` (`htable *ht`, int no\_destroy)
- `htable * cluster_sort` (int col1, int col2, `htable *ht`, int no\_destroy, int sort)
- int \* `col_names_to_vars` (`llist *varnames`)
- `htable * load_from_tuples` (int id, char \*cons\_name, char \*\*col\_names, tuple \*tuples, int \*vars, int header\_only, `constraint *cons`)
- `htable * load_from_conflict_tuples` (int id, char \*cons\_name, char \*\*col\_names, tuple \*tuples, `llist *cons_var_domains`, int \*vars)
- `htable * convert_conflict_to_support4` (`htable *conf`)
- `htable * convert_conflict_to_supportn` (`htable *conf`)
- int \* `init_get_next_tuple` (`htable *ht`)
- int `get_next_tuple` (`htable *ht`, int \*current\_tuple)
- void `convert_conflict_to_supportn_rec` (`htable *conf`, `htable *support`, int \*current\_tuple\_pos, int \*t, int v)
- `htable * convert_conflict_to_support3` (`htable *conf`)
- `htable * convert_conflict_to_support2` (`htable *conf`)
- int `get_int_common_col_count` (int \*var1, int \*var2, int \*vmap)
- int \* `get_projection_scope` (`llist *var1`, int \*var2, int \*vmap)



- void `destroy_int_common_cols_list` (`llist *`)
- `llist *` `get_int_common_cols_list` (`llist *var1`, `int *var2`, `int *vmap`)
- `llist *` `get_int_common_cols` (`int *var1`, `int *var2`, `int *vmap`)
- `llist *` `get_int_common_cols_vars_list` (`int *var1`, `int *var2`, `llist *varList`, `int *vmap`)
- `int` `has_int_common_cols` (`int *var1`, `int *var2`, `int *vmap`)
- `common_var *` `new_common_var` (`int a`, `int b`)
- `int` `do_compare` (`htable *hta`, `htable *htb`, `llist *common`, `int i`, `int j`)
- `int` `do_match` (`htable *hta`, `htable *htb`, `llist *common`, `int i`, `int j`)
- `int` `do_match_next` (`htable *hta`, `llist *common`, `int i`, `int hta_htb`)
- void `merge_tuple` (`htable *hta`, `htable *htb`, `int *tc`, `int *ta`, `int *tb`, `int *common_map_htb`)
- void `merge` (`htable *hta`, `htable *htb`, `int *tc`, `int x`, `int y`, `int *common_map_htb`)
- `int` `compare_clustered` (`void *ba`, `void *bb`)
- `htable *` `hash_join_tables` (`htable *hta`, `htable *htb`)
- `htable *` `project_join_tables` (`htable *hta`, `htable *htb`)
- `htable *` `join_tables_new` (`htable *htap`, `htable *htbp`)
- `htable *` `join_tables` (`htable **htap`, `htable **htbp`, `int limit`, `htable *joined_table`, `int no_destroy`)
- `htable *` `join_tables_w_suspend` (`htable **htap`, `htable **htbp`, `int limit`, `htable *joined_table`, `int no_destroy`)
- `llist *` `get_common_cols` (`htable *hta`, `htable *htb`)
- `int` `match_tuple_a_tuple_b` (`int *a`, `int *b`, `llist *common_var_array_list`)
- `htable *` `distinct` (`htable *ht`)
- `htable *` `project_table` (`htable *ht`, `llist *cols`)
- `htable *` `project_distinct_table` (`htable *ht`, `llist *cols`, `int distinct`)
- `htable *` `purge_htable` (`htable *ht`)
- `htable *` `copy_htable` (`htable *ht`)
- `hashmap *` `init_hashmap` (`htable *ht`, `llist *cols`)
- `hashmap *` `get_hashmap_wd` (`htable *ht`, `llist *cols`)
- `htable *` `hash_project_distinct_table` (`htable *ht`, `llist *cols`, `int distinct`)
- `htable *` `prepare_pipelined_join` (`htable **htap`, `htable **htbp`, `htable *joined_table`, `int no_destroy`)
- void `filter_table` (`htable *ht`, `alive_table *at`)
- void `update_alive` (`alive_table *at`, `htable *ht`)
- void `track_alive` (`htable *ht`)
- void `delete_tuple` (`htable *ht`, `int tuple`)
- void `un_delete_tuple` (`htable *ht`, `int tuple`)
- void `un_delete_tuple_old` (`htable *ht`, `int tuple`)
- `int` `match_tuple_single_ht` (`int *tuple`, `llist *commons`, `int twisted`, `htable *ht`)
- `int` `get_next_alive_index` (`int t`, `htable *ht`)
- `int *` `get_next` (`htable *ht`)
- `int` `get_it_value` (`void *a`, `int i`, `int c`)
- `void *` `get_it_object` (`void *a`, `int i`, `int c`)
- void `set_it_object` (`void *a`, `int i`, `int c`, `void *obj`)
- void `sort_ht_iterator` (`htable *ht`)
- `tuple_ids *` `new_tuple_ids` (`int *tuple`)
- `tuple_ids *` `merge_two_tuples_ids` (`tuple_ids *ta`, `tuple_ids *tb`, `llist *comms`)
- void `free_tuple_ids` (`tuple_ids *ti`)
- `htable *` `create_fill_htable` (`llist *varptrs`, `const char *name`)
- void `update_current_domains` (`htable *ht`, `light_stack *undo_value_stack`)
- void `filter_table_with_current_domains` (`htable *ht`)
- double `get_domain_product` (`htable *ht`)
- void `resetset_variable_hashtables` (`llist *htables`)
- void `set_variable_hashtables` (`llist *htables`)
- void `set_variable_hashtable` (`htable *ht`)
- void `filter_domains_with_unary_constraints` (`variables *vars`)
- `int` `compare_scope` (`const void *o1`, `const void *o2`)
- `int` `compare_scope_buble` (`void *o1`, `void *o2`)

## Variables

- int `current_markgeneration` = 0

### 4.27.1 Function Documentation

4.27.1.1 void `add_name` ( char \* *name*, char \*\* *names*, int *pos* )

4.27.1.2 `htable*` `add_tuple` ( int \* *tuple*, `htable` \* *ht* )

4.27.1.3 void `add_tuple_list` ( `htable` \* *ht* )

4.27.1.4 void `build_histogram` ( `htable` \* *table* )

4.27.1.5 double `ceil` ( double *x* )

4.27.1.6 void `clear_deleted_rows` ( `htable` \* *ht* )

4.27.1.7 `htable*` `cluster` ( int *col1*, int *col2*, `htable` \* *ht*, int *no\_destroy* )

4.27.1.8 `htable*` `cluster_nosort` ( `htable` \* *ht*, int *no\_destroy* )

4.27.1.9 `htable*` `cluster_sort` ( int *col1*, int *col2*, `htable` \* *ht*, int *no\_destroy*, int *sort* )

4.27.1.10 int\* `col_names_to_vars` ( `llist` \* *varnames* )

4.27.1.11 int `compare_clustered` ( void \* *ba*, void \* *bb* )

4.27.1.12 int `compare_col_card` ( void \* *value*, void \* *bodyb* )

4.27.1.13 int `compare_col_name` ( void \* *value*, void \* *bodyb* )

4.27.1.14 int `compare_scope` ( const void \* *o1*, const void \* *o2* )

4.27.1.15 int `compare_scope_buble` ( void \* *o1*, void \* *o2* )

4.27.1.16 `htable*` `convert_conflict_to_support2` ( `htable` \* *conf* )

4.27.1.17 `htable*` `convert_conflict_to_support3` ( `htable` \* *conf* )

4.27.1.18 `htable*` `convert_conflict_to_support4` ( `htable` \* *conf* )

4.27.1.19 `htable*` `convert_conflict_to_supportn` ( `htable` \* *conf* )

4.27.1.20 void `convert_conflict_to_supportn_rec` ( `htable` \* *conf*, `htable` \* *support*, int \* *current\_tuple\_pos*, int \* *t*, int *v* )

4.27.1.21 char\*\* `copy_col_names` ( char \*\* *col\_names*, int *size* )

4.27.1.22 `htable*` `copy_htable` ( `htable` \* *ht* )

4.27.1.23 `htable*` `create_fill_htable` ( `llist` \* *varptrs*, const char \* *name* )

4.27.1.24 void `delete_tuple` ( `htable` \* *ht*, int *tuple* )

4.27.1.25 void `destroy_histogram` ( `histogram` \* *histo* )

- 4.27.1.26 void destroy\_hhtable ( htable \* ht )
- 4.27.1.27 void destroy\_hhtable\_o ( htable \* ht, int kill\_col\_names )
- 4.27.1.28 void destroy\_int\_common\_cols\_list ( llist \* l )
- 4.27.1.29 htable\* distinct ( htable \* ht )
- 4.27.1.30 int do\_compare ( htable \* hta, htable \* htb, llist \* common, int i, int j )
- 4.27.1.31 int do\_match ( htable \* hta, htable \* htb, llist \* common, int i, int j )
- 4.27.1.32 int do\_match\_next ( htable \* hta, llist \* common, int i, int hta\_htb )
- 4.27.1.33 int estimate\_join\_project\_size ( htable \* hta, htable \* htb, int project )
- 4.27.1.34 int estimate\_join\_size ( htable \* hta, htable \* htb )
- 4.27.1.35 int estimate\_join\_size\_noproject ( htable \* hta, htable \* htb )
- 4.27.1.36 int estimate\_project\_size ( htable \* hta, llist \* project\_cols )
- 4.27.1.37 void filter\_domains\_with\_unary\_constraints ( variables \* vars )
- 4.27.1.38 void filter\_table ( htable \* ht, alive\_table \* at )
- 4.27.1.39 void filter\_table\_with\_current\_domains ( htable \* ht )
- 4.27.1.40 void fprintf\_ht ( FILE \* file, htable \* ht, int short\_name )
- 4.27.1.41 void free\_tuple\_ids ( tuple\_ids \* ti )
- 4.27.1.42 llist\* get\_common\_cols ( htable \* hta, htable \* htb )
- 4.27.1.43 double get\_domain\_product ( htable \* ht )
- 4.27.1.44 hashmap\* get\_hashmap\_wd ( htable \* ht, llist \* cols )
- 4.27.1.45 llist\* get\_ht\_iterator ( htable \* ht )
- 4.27.1.46 int get\_int\_common\_col\_count ( int \* var1, int \* var2, int \* vmap )
- 4.27.1.47 llist\* get\_int\_common\_cols ( int \* var1, int \* var2, int \* vmap )
- 4.27.1.48 llist\* get\_int\_common\_cols\_list ( llist \* var1, int \* var2, int \* vmap )
- 4.27.1.49 llist\* get\_int\_common\_cols\_vars\_list ( int \* var1, int \* var2, llist \* varList, int \* vmap )
- 4.27.1.50 int\* get\_int\_pair ( void )
- 4.27.1.51 void\* get\_it\_object ( void \* a, int i, int c )
- 4.27.1.52 int get\_it\_value ( void \* a, int i, int c )
- 4.27.1.53 int\* get\_next ( htable \* ht )

- 4.27.1.54 `int get_next_alive_index ( int t, htable * ht )`
- 4.27.1.55 `int get_next_tuple ( htable * ht, int * current_tuple )`
- 4.27.1.56 `int* get_projection_scope ( llist * var1, int * var2, int * vmap )`
- 4.27.1.57 `int* get_tuple ( int row, htable * ht )`
- 4.27.1.58 `double get_tuple_value ( int row, htable * ht )`
- 4.27.1.59 `int get_value ( int row, int column, htable * ht )`
- 4.27.1.60 `int get_value_sort ( void * ht, int row, int column )`
- 4.27.1.61 `int has_int_common_cols ( int * var1, int * var2, int * vmap )`
- 4.27.1.62 `htable* hash_join_tables ( htable * hta, htable * htb )`
- 4.27.1.63 `htable* hash_project_distinct_table ( htable * ht, llist * cols, int distinct )`
- 4.27.1.64 `int have_value ( int x )`
- 4.27.1.65 `int* init_get_next_tuple ( htable * ht )`
- 4.27.1.66 `hashmap* init_hashmap ( htable * ht, llist * cols )`
- 4.27.1.67 `htable* join_tables ( htable ** htap, htable ** htbp, int limit, htable * joined_table, int no_destroy )`
- 4.27.1.68 `htable* join_tables_new ( htable * htap, htable * htbp )`
- 4.27.1.69 `htable* join_tables_w_suspend ( htable ** htap, htable ** htbp, int limit, htable * joined_table, int no_destroy )`
- 4.27.1.70 `htable* load_from_conflict_tuples ( int id, char * cons_name, char ** col_names, tuple * tuples, llist * cons_var_domains, int * vars )`
- 4.27.1.71 `htable* load_from_tuples ( int id, char * cons_name, char ** col_names, tuple * tuples, int * vars, int header_only, constraint * cons )`
- 4.27.1.72 `int match_tuple_a_tuple_b ( int * a, int * b, llist * common_var_array_list )`
- 4.27.1.73 `int match_tuple_single_ht ( int * tuple, llist * commons, int twisted, htable * ht )`
- 4.27.1.74 `void merge ( htable * hta, htable * htb, int * tc, int x, int y, int * common_map_htb )`
- 4.27.1.75 `void merge_tuple ( htable * hta, htable * htb, int * tc, int * ta, int * tb, int * common_map_htb )`
- 4.27.1.76 `tuple_ids* merge_two_tuples_ids ( tuple_ids * ta, tuple_ids * tb, llist * comms )`
- 4.27.1.77 `htable* new_clustered_htable ( int id, char * name, int width, int height, char ** col_names, int join_cutoff, int * vars )`
- 4.27.1.78 `common_var* new_common_var ( int a, int b )`
- 4.27.1.79 `histogram* new_histogram ( int columns )`

- 4.27.1.80 `htable* new_hhtable ( int id, const char * name, int width, int height, htable * prev, char ** col_names, int join_cutoff, int is_first, int * vars )`
- 4.27.1.81 `htable* new_hhtable_replace ( int id, const char * name, int width, int height, htable * prev, char ** col_names, int join_cutoff, int is_first, int * vars, int replace )`
- 4.27.1.82 `char** new_names ( int size )`
- 4.27.1.83 `tuple_ids* new_tuple_ids ( int * tuple )`
- 4.27.1.84 `htable* prepare_pipelined_join ( htable ** htap, htable ** htbp, htable * joined_table, int no_destroy )`
- 4.27.1.85 `void print_ht ( htable * ht, int short_name )`
- 4.27.1.86 `void print_ht_h ( htable * ht, int short_name )`
- 4.27.1.87 `void print_ht_line ( htable * ht )`
- 4.27.1.88 `void print_ht_mysql ( htable * ht )`
- 4.27.1.89 `htable* project_distinct_table ( htable * ht, llist * cols, int distinct )`
- 4.27.1.90 `htable* project_join_tables ( htable * hta, htable * htb )`
- 4.27.1.91 `htable* project_table ( htable * ht, llist * cols )`
- 4.27.1.92 `htable* purge_hhtable ( htable * ht )`
- 4.27.1.93 `void rebuild_histogram ( htable * table )`
- 4.27.1.94 `void replace_data ( htable * data, htable * ht )`
- 4.27.1.95 `void resetset_variable_hashtables ( llist * htables )`
- 4.27.1.96 `void set_it_object ( void * a, int i, int c, void * obj )`
- 4.27.1.97 `void set_tuple_value ( double value, int row, htable * ht )`
- 4.27.1.98 `void set_value_sort ( void * vht, int row, int column, int value )`
- 4.27.1.99 `void set_variable_hashtable ( htable * ht )`
- 4.27.1.100 `void set_variable_hashtables ( llist * htables )`
- 4.27.1.101 `void sort_ht_iterator ( htable * ht )`
- 4.27.1.102 `void track_alive ( htable * ht )`
- 4.27.1.103 `void un_delete_tuple ( htable * ht, int tuple )`
- 4.27.1.104 `void un_delete_tuple_old ( htable * ht, int tuple )`
- 4.27.1.105 `void update_alive ( alive_table * at, htable * ht )`
- 4.27.1.106 `void update_current_domains ( htable * ht, light_stack * undo_value_stack )`

4.27.1.107 void update\_histogram ( htable \* table )

## 4.27.2 Variable Documentation

4.27.2.1 int current\_markgeneration = 0

## 4.28 hashtable.h File Reference

```
#include <stdio.h>
#include "relations.h"
#include "variable_tracking.h"
#include "constraints.h"
#include "hashmap.h"
#include "variables.h"
#include "globals.h"
```

### Classes

- struct [htable](#)
- struct [histogram](#)
- struct [common\\_var](#)
- struct [col\\_card](#)
- struct [tuple\\_ids](#)

### Macros

- #define [DEF\\_HIGHT](#) 4
- #define [INF\\_HIGHT](#) 999999999
- #define [DELETED](#) width+1
- #define [TIMESTP](#) width+2
- #define [REVTIME](#) width+3
- #define [TUPMARK](#) width+4
- #define [ADD\\_FOR\\_PHYSICAL\\_WIDTH](#) 6

### Typedefs

- typedef struct [htable](#) [htable](#)
- typedef struct [histogram](#) [histogram](#)
- typedef struct [common\\_var](#) [common\\_var](#)
- typedef struct [col\\_card](#) [col\\_card](#)
- typedef struct [tuple\\_ids](#) [tuple\\_ids](#)

### Functions

- int \* [get\\_int\\_pair](#) (void)
- int [compare\\_col\\_card](#) (void \*value, void \*bodyb)
- int [compare\\_col\\_name](#) (void \*value, void \*bodyb)
- [histogram](#) \* [new\\_histogram](#) (int columns)
- void [build\\_histogram](#) ([htable](#) \*table)
- void [rebuild\\_histogram](#) ([htable](#) \*table)
- void [update\\_histogram](#) ([htable](#) \*table)

- int `estimate_join_size_noproject` (`htable *hta`, `htable *htb`)
- int `estimate_project_size` (`htable *hta`, `llist *project_cols`)
- int `estimate_join_project_size` (`htable *hta`, `htable *htb`, int `project`)
- `htable * hash_project_distinct_table` (`htable *ht`, `llist *cols`, int `distinct`)
- int `estimate_join_size` (`htable *ha`, `htable *hb`)
- `htable * new_htable_replace` (int `id`, const char \*`name`, int `width`, int `hight`, `htable *prev`, char \*\*`col_names`, int `join_cutoff`, int `is_first`, int \*`vars`, int `replace`)
- `htable * new_htable` (int `id`, const char \*`name`, int `width`, int `hight`, `htable *prev`, char \*\*`col_names`, int `join_cutoff`, int `is_first`, int \*`vars`)
- char \*\* `copy_col_names` (char \*\*`col_names`, int `size`)
- `htable * new_clustered_htable` (int `id`, char \*`name`, int `width`, int `hight`, char \*\*`col_names`, int `join_cutoff`, int \*`vars`)
- int \* `get_tuple` (int `row`, `htable *ht`)
- double `get_tuple_value` (int `row`, `htable *ht`)
- void `set_tuple_value` (double `value`, int `row`, `htable *ht`)
- int `get_value` (int `row`, int `column`, `htable *ht`)
- void `destroy_htable` (`htable *ht`)
- void `destroy_htable_o` (`htable *ht`, int `kill_col_names`)
- char \*\* `new_names` (int `size`)
- void `add_name` (char \*`name`, char \*\*`names`, int `pos`)
- `htable * add_tuple` (int \*`tuple`, `htable *ht`)
- `htable * cluster_nosort` (`htable *ht`, int `no_destroy`)
- `htable * cluster_sort` (int `col1`, int `col2`, `htable *ht`, int `no_destroy`, int `sort`)
- `htable * cluster` (int `col1`, int `col2`, `htable *ht`, int `no_destroy`)
- void `print_ht_h` (`htable *ht`, int `short_name`)
- void `print_ht` (`htable *ht`, int `short_name`)
- void `fprint_ht` (FILE \*`file`, `htable *ht`, int `short_name`)
- void `print_ht_mysql` (`htable *ht`)
- `htable * load_from_tuples` (int `id`, char \*`cons_name`, char \*\*`col_names`, tuple \*`tuples`, int \*`vars`, int `header_only`, `constraint *cons`)
- `htable * load_from_conflict_tuples` (int `id`, char \*`cons_name`, char \*\*`col_names`, tuple \*`tuples`, `llist *cons_var_domains`, int \*`vars`)
- `htable * convert_conflict_to_support2` (`htable *conf`)
- `htable * convert_conflict_to_support3` (`htable *conf`)
- `htable * convert_conflict_to_support4` (`htable *conf`)
- `htable * project_distinct_table` (`htable *ht`, `llist *cols`, int `distinct`)
- void `replace_data` (`htable *data`, `htable *ht`)
- `htable * project_table` (`htable *ht`, `llist *cols`)
- `htable * project_join_tables` (`htable *hta`, `htable *htb`)
- `htable * join_tables` (`htable **htap`, `htable **htbp`, int `limit`, `htable *joined_table`, int `no_destroy`)
- `htable * hash_join_tables` (`htable *hta`, `htable *htb`)
- `htable * join_tables_w_suspend` (`htable **htap`, `htable **htbp`, int `limit`, `htable *joined_table`, int `no_destroy`)
- void `destroy_int_common_cols_list` (`llist *`)
- `llist * get_common_cols` (`htable *hta`, `htable *htb`)
- `llist * get_int_common_cols_list` (`llist *var1`, int \*`var2`, int \*`vmap`)
- `llist * get_int_common_cols_vars_list` (int \*`var1`, int \*`var2`, `llist *varList`, int \*`vmap`)
- int \* `get_projection_scope` (`llist *var1`, int \*`var2`, int \*`vmap`)
- `htable * distinct` (`htable *ht`)
- int `get_value_sort` (void \*`ht`, int `row`, int `column`)
- void `set_value_sort` (void \*`vht`, int `row`, int `column`, int `value`)
- int `have_value` (int `x`)
- int `get_int_common_col_count` (int \*`var1`, int \*`var2`, int \*`vmap`)
- int `has_int_common_cols` (int \*`var1`, int \*`var2`, int \*`vmap`)
- `llist * get_int_common_cols` (int \*`var1`, int \*`var2`, int \*`vmap`)
- `common_var * new_common_var` (int `a`, int `b`)

- `int do_compare (htable *hta, htable *htb, llist *common, int i, int j)`
- `int do_match (htable *hta, htable *htb, llist *common, int i, int j)`
- `int do_match_next (htable *hta, llist *common, int i, int hta_htb)`
- `int match_tuple_a_tuple_b (int *a, int *b, llist *common_var_array_list)`
- `void merge (htable *hta, htable *htb, int *tc, int x, int y, int *common_map_htb)`
- `void merge_tuple (htable *hta, htable *htb, int *tc, int *ta, int *tb, int *common_map_htb)`
- `int compare_clustered (void *ba, void *bb)`
- `htable * prepare_pipelined_join (htable **htap, htable **htbp, htable *joined_table, int no_destroy)`
- `hashmap * get_hashmap_wd (htable *ht, llist *cols)`
- `hashmap * init_hashmap (htable *ht, llist *cols)`
- `void update_alive (alive_table *at, htable *ht)`
- `void filter_table (htable *ht, alive_table *at)`
- `void track_alive (htable *ht)`
- `void clear_deleted_rows (htable *ht)`
- `void destroy_histogram (histogram *histo)`
- `void delete_tuple (htable *ht, int tuple)`
- `void un_delete_tuple_old (htable *ht, int tuple)`
- `void un_delete_tuple (htable *ht, int tuple)`
- `int match_tuple_single_ht (int *tuple, llist *commons, int twisted, htable *ht)`
- `int * get_next (htable *ht)`
- `void add_tuple_list (htable *ht)`
- `llist * get_ht_iterator (htable *ht)`
- `int get_it_value (void *a, int i, int c)`
- `void * get_it_object (void *a, int i, int c)`
- `void set_it_object (void *a, int i, int c, void *obj)`
- `void sort_ht_iterator (htable *ht)`
- `int get_next_alive_index (int t, htable *ht)`
- `int * col_names_to_vars (llist *varnames)`
- `tuple_ids * new_tuple_ids (int *tuple)`
- `void free_tuple_ids (tuple_ids *ti)`
- `tuple_ids * merge_two_tuples_ids (tuple_ids *ta, tuple_ids *tb, llist *comms)`
- `void convert_conflict_to_supportn_rec (htable *conf, htable *support, int *current_tuple, int *t, int v)`
- `htable * convert_conflict_to_supportn (htable *conf)`
- `htable * create_fill_htable (llist *varptrs, const char *name)`
- `int * init_get_next_tuple (htable *ht)`
- `int get_next_tuple (htable *ht, int *current_tuple)`
- `void update_current_domains (htable *ht, light_stack *undo_value_stack)`
- `void filter_table_with_current_domains (htable *ht)`
- `htable * join_tables_new (htable *htap, htable *htbp)`
- `double get_domain_product (htable *ht)`
- `htable * copy_htable (htable *ht)`
- `htable * purge_htable (htable *ht)`
- `void print_ht_line (htable *ht)`
- `void set_variable_hashtables (llist *htables)`
- `void set_variable_hashtable (htable *ht)`
- `void resetset_variable_hashtables (llist *htables)`
- `void filter_domains_with_unary_constraints (variables *vars)`
- `int compare_scope (const void *o1, const void *o2)`
- `int compare_scope_buble (void *o1, void *o2)`

#### 4.28.1 Detailed Description

Provides the data structures for representing the relations of the constraints and methods for constructing and manipulating them.



## 4.28.2 Macro Definition Documentation

4.28.2.1 `#define ADD_FOR_PHYSICAL_WIDTH 6`

4.28.2.2 `#define DEF_HIGHT 4`

4.28.2.3 `#define DELETED width+1`

4.28.2.4 `#define INF_HIGHT 999999999`

4.28.2.5 `#define REVTIME width+3`

4.28.2.6 `#define Timestp width+2`

4.28.2.7 `#define TUPMARK width+4`

## 4.28.3 Typedef Documentation

4.28.3.1 `typedef struct col_card col_card`

4.28.3.2 `typedef struct common_var common_var`

4.28.3.3 `typedef struct histogram histogram`

4.28.3.4 `typedef struct htable htable`

4.28.3.5 `typedef struct tuple_ids tuple_ids`

## 4.28.4 Function Documentation

4.28.4.1 `void add_name ( char * name, char ** names, int pos )`

4.28.4.2 `htable* add_tuple ( int * tuple, htable * ht )`

4.28.4.3 `void add_tuple_list ( htable * ht )`

4.28.4.4 `void build_histogram ( htable * table )`

4.28.4.5 `void clear_deleted_rows ( htable * ht )`

4.28.4.6 `htable* cluster ( int col1, int col2, htable * ht, int no_destroy )`

4.28.4.7 `htable* cluster_nosort ( htable * ht, int no_destroy )`

4.28.4.8 `htable* cluster_sort ( int col1, int col2, htable * ht, int no_destroy, int sort )`

4.28.4.9 `int* col_names_to_vars ( llist * varnames )`

4.28.4.10 `int compare_clustered ( void * ba, void * bb )`

4.28.4.11 `int compare_col_card ( void * value, void * bodyb )`

4.28.4.12 `int compare_col_name ( void * value, void * bodyb )`

4.28.4.13 `int compare_scope ( const void * o1, const void * o2 )`

- 4.28.4.14 `int compare_scope_buble ( void * o1, void * o2 )`
- 4.28.4.15 `htable* convert_conflict_to_support2 ( htable * conf )`
- 4.28.4.16 `htable* convert_conflict_to_support3 ( htable * conf )`
- 4.28.4.17 `htable* convert_conflict_to_support4 ( htable * conf )`
- 4.28.4.18 `htable* convert_conflict_to_supportn ( htable * conf )`
- 4.28.4.19 `void convert_conflict_to_supportn_rec ( htable * conf, htable * support, int * current_tuple, int * t, int v )`
- 4.28.4.20 `char** copy_col_names ( char ** col_names, int size )`
- 4.28.4.21 `htable* copy_hhtable ( htable * ht )`
- 4.28.4.22 `htable* create_fill_hhtable ( llist * varptrs, const char * name )`
- 4.28.4.23 `void delete_tuple ( htable * ht, int tuple )`
- 4.28.4.24 `void destroy_histogram ( histogram * histo )`
- 4.28.4.25 `void destroy_hhtable ( htable * ht )`
- 4.28.4.26 `void destroy_hhtable_o ( htable * ht, int kill_col_names )`
- 4.28.4.27 `void destroy_int_common_cols_list ( llist * l )`
- 4.28.4.28 `htable* distinct ( htable * ht )`
- 4.28.4.29 `int do_compare ( htable * hta, htable * htb, llist * common, int i, int j )`
- 4.28.4.30 `int do_match ( htable * hta, htable * htb, llist * common, int i, int j )`
- 4.28.4.31 `int do_match_next ( htable * hta, llist * common, int i, int hta_htb )`
- 4.28.4.32 `int estimate_join_project_size ( htable * hta, htable * htb, int project )`
- 4.28.4.33 `int estimate_join_size ( htable * ha, htable * hb )`
- 4.28.4.34 `int estimate_join_size_noproject ( htable * hta, htable * htb )`
- 4.28.4.35 `int estimate_project_size ( htable * hta, llist * project_cols )`
- 4.28.4.36 `void filter_domains_with_unary_constraints ( variables * vars )`
- 4.28.4.37 `void filter_table ( htable * ht, alive_table * at )`
- 4.28.4.38 `void filter_table_with_current_domains ( htable * ht )`
- 4.28.4.39 `void fprintf_ht ( FILE * file, htable * ht, int short_name )`
- 4.28.4.40 `void free_tuple_ids ( tuple_ids * ti )`
- 4.28.4.41 `llist* get_common_cols ( htable * hta, htable * htb )`

- 4.28.4.42 `double get_domain_product ( htable * ht )`
- 4.28.4.43 `hashmap* get_hashmap_wd ( htable * ht, llist * cols )`
- 4.28.4.44 `llist* get_ht_iterator ( htable * ht )`
- 4.28.4.45 `int get_int_common_col_count ( int * var1, int * var2, int * vmap )`
- 4.28.4.46 `llist* get_int_common_cols ( int * var1, int * var2, int * vmap )`
- 4.28.4.47 `llist* get_int_common_cols_list ( llist * var1, int * var2, int * vmap )`
- 4.28.4.48 `llist* get_int_common_cols_vars_list ( int * var1, int * var2, llist * varList, int * vmap )`
- 4.28.4.49 `int* get_int_pair ( void )`
- 4.28.4.50 `void* get_it_object ( void * a, int i, int c )`
- 4.28.4.51 `int get_it_value ( void * a, int i, int c )`
- 4.28.4.52 `int* get_next ( htable * ht )`
- 4.28.4.53 `int get_next_alive_index ( int t, htable * ht )`
- 4.28.4.54 `int get_next_tuple ( htable * ht, int * current_tuple )`
- 4.28.4.55 `int* get_projection_scope ( llist * var1, int * var2, int * vmap )`
- 4.28.4.56 `int* get_tuple ( int row, htable * ht )`
- 4.28.4.57 `double get_tuple_value ( int row, htable * ht )`
- 4.28.4.58 `int get_value ( int row, int column, htable * ht )`
- 4.28.4.59 `int get_value_sort ( void * ht, int row, int column )`
- 4.28.4.60 `int has_int_common_cols ( int * var1, int * var2, int * vmap )`
- 4.28.4.61 `htable* hash_join_tables ( htable * hta, htable * htb )`
- 4.28.4.62 `htable* hash_project_distinct_table ( htable * ht, llist * cols, int distinct )`
- 4.28.4.63 `int have_value ( int x )`
- 4.28.4.64 `int* init_get_next_tuple ( htable * ht )`
- 4.28.4.65 `hashmap* init_hashmap ( htable * ht, llist * cols )`
- 4.28.4.66 `htable* join_tables ( htable ** htap, htable ** htbp, int limit, htable * joined_table, int no_destroy )`
- 4.28.4.67 `htable* join_tables_new ( htable * htap, htable * htbp )`
- 4.28.4.68 `htable* join_tables_w_suspend ( htable ** htap, htable ** htbp, int limit, htable * joined_table, int no_destroy )`

- 4.28.4.69 `htable*` `load_from_conflict_tuples` ( `int` *id*, `char *` *cons\_name*, `char **` *col\_names*, `tuple *` *tuples*, `llist *` *cons\_var\_domains*, `int *` *vars* )
- 4.28.4.70 `htable*` `load_from_tuples` ( `int` *id*, `char *` *cons\_name*, `char **` *col\_names*, `tuple *` *tuples*, `int *` *vars*, `int` *header\_only*, `constraint *` *cons* )
- 4.28.4.71 `int` `match_tuple_a_tuple_b` ( `int *` *a*, `int *` *b*, `llist *` *common\_var\_array\_list* )
- 4.28.4.72 `int` `match_tuple_single_ht` ( `int *` *tuple*, `llist *` *commons*, `int` *twisted*, `htable *` *ht* )
- 4.28.4.73 `void` `merge` ( `htable *` *hta*, `htable *` *htb*, `int *` *tc*, `int` *x*, `int` *y*, `int *` *common\_map\_htb* )
- 4.28.4.74 `void` `merge_tuple` ( `htable *` *hta*, `htable *` *htb*, `int *` *tc*, `int *` *ta*, `int *` *tb*, `int *` *common\_map\_htb* )
- 4.28.4.75 `tuple_ids*` `merge_two_tuples_ids` ( `tuple_ids *` *ta*, `tuple_ids *` *tb*, `llist *` *comms* )
- 4.28.4.76 `htable*` `new_clustered_htable` ( `int` *id*, `char *` *name*, `int` *width*, `int` *hight*, `char **` *col\_names*, `int` *join\_cutoff*, `int *` *vars* )
- 4.28.4.77 `common_var*` `new_common_var` ( `int` *a*, `int` *b* )
- 4.28.4.78 `histogram*` `new_histogram` ( `int` *columns* )
- 4.28.4.79 `htable*` `new_htable` ( `int` *id*, `const char *` *name*, `int` *width*, `int` *hight*, `htable *` *prev*, `char **` *col\_names*, `int` *join\_cutoff*, `int` *is\_first*, `int *` *vars* )
- 4.28.4.80 `htable*` `new_htable_replace` ( `int` *id*, `const char *` *name*, `int` *width*, `int` *hight*, `htable *` *prev*, `char **` *col\_names*, `int` *join\_cutoff*, `int` *is\_first*, `int *` *vars*, `int` *replace* )
- 4.28.4.81 `char**` `new_names` ( `int` *size* )
- 4.28.4.82 `tuple_ids*` `new_tuple_ids` ( `int *` *tuple* )
- 4.28.4.83 `htable*` `prepare_pipelined_join` ( `htable **` *htap*, `htable **` *htbp*, `htable *` *joined\_table*, `int` *no\_destroy* )
- 4.28.4.84 `void` `print_ht` ( `htable *` *ht*, `int` *short\_name* )
- 4.28.4.85 `void` `print_ht_h` ( `htable *` *ht*, `int` *short\_name* )
- 4.28.4.86 `void` `print_ht_line` ( `htable *` *ht* )
- 4.28.4.87 `void` `print_ht_mysql` ( `htable *` *ht* )
- 4.28.4.88 `htable*` `project_distinct_table` ( `htable *` *ht*, `llist *` *cols*, `int` *distinct* )
- 4.28.4.89 `htable*` `project_join_tables` ( `htable *` *hta*, `htable *` *htb* )
- 4.28.4.90 `htable*` `project_table` ( `htable *` *ht*, `llist *` *cols* )
- 4.28.4.91 `htable*` `purge_htable` ( `htable *` *ht* )
- 4.28.4.92 `void` `rebuild_histogram` ( `htable *` *table* )
- 4.28.4.93 `void` `replace_data` ( `htable *` *data*, `htable *` *ht* )
- 4.28.4.94 `void` `resetset_variable_hashtables` ( `llist *` *htables* )

- 4.28.4.95 void `set_it_object` ( void \* *a*, int *i*, int *c*, void \* *obj* )
- 4.28.4.96 void `set_tuple_value` ( double *value*, int *row*, htable \* *ht* )
- 4.28.4.97 void `set_value_sort` ( void \* *vht*, int *row*, int *column*, int *value* )
- 4.28.4.98 void `set_variable_hashtable` ( htable \* *ht* )
- 4.28.4.99 void `set_variable_hashtables` ( llist \* *htables* )
- 4.28.4.100 void `sort_ht_iterator` ( htable \* *ht* )
- 4.28.4.101 void `track_alive` ( htable \* *ht* )
- 4.28.4.102 void `un_delete_tuple` ( htable \* *ht*, int *tuple* )
- 4.28.4.103 void `un_delete_tuple_old` ( htable \* *ht*, int *tuple* )
- 4.28.4.104 void `update_alive` ( alive\_table \* *at*, htable \* *ht* )
- 4.28.4.105 void `update_current_domains` ( htable \* *ht*, light\_stack \* *undo\_value\_stack* )
- 4.28.4.106 void `update_histogram` ( htable \* *table* )

## 4.29 hashtable\_search.c File Reference

```
#include "hashtable.h"
#include "hashtable_search.h"
#include "utils.h"
#include "active_tuple.h"
#include "rs2.h"
#include "decomposition.h"
#include "splitter.h"
#include "llist.h"
#include "sort.h"
```

### Functions

- htable \* `get_projected_on_variables` (htable \**ht*, llist \**variablelist*)
- htable \* `join_filter_relations_with_all_sols` (llist \**htablelist*, int *nojoin*, int *nofilter*, char \**optionalname*)
- htable \* `project_with_tree` (htable \**ht*, llist \**varlist*)
- htable \* `project_with_tree_payload` (htable \**ht*, llist \**varlist*, tree\_map \*\**tree\_only*, void \**get\_payload*(int *param*), int *param*)
- void `sort_with_tree` (htable \**ht*, llist \**varlist*, int \**vmap*)
- llist \* `project_combine_relations` (llist \**htables*, llist \**varlist*, int *filter*)
- llist \* `generate_tables_for_scopes_using_relations` (llist \**list\_of\_lvids*, llist \**using\_rels*, int *id*, int *scopes\_int\_p*, int *record\_stats*)
- llist \* `generate_tables_for_scopes_using_relations_noproject` (llist \**list\_of\_lvids*, llist \**using\_rels*, int *id*, int *scopes\_int\_p*, int *record\_stats*)
- llist \* `join_project_multiple_in_multiple_out` (llist \**htablelist*, llist \**list\_of\_domain\_lists*, int *nofilteroriginal*, int *id*)
- int `filter_this_relation_with_1_sol` (htable \**htfilter*, llist \**given\_htablelist*, set \**undo\_set*, light\_stack \**undo\_stack*)
- int `filter_relations_with_1_sol` (llist \**htablelist*)

- `llist * join_project_multiple_in_multiple_out_noextend (llist *htablelist, llist *list_of_domain_lists, int id)`
- `llist * join_project_multiple_in_multiple_out_noextend_given_tables (llist *first_tables, llist *htablelist_ofrest, llist *list_of_domain_lists, int id)`
- `int process_with_all_sols (llist *htablelist, active_tuple *ac, options category)`
- `int process_with_all_sols_first_var (llist *htablelist, active_tuple *ac, int first_v, options category)`

#### 4.29.1 Function Documentation

4.29.1.1 `int filter_relations_with_1_sol ( llist * htablelist )`

4.29.1.2 `int filter_this_relation_with_1_sol ( htable * hfilter, llist * given_htablelist, set * undo_set, light_stack * undo_stack )`

4.29.1.3 `llist* generate_tables_for_scopes_using_relations ( llist * list_of_lvids, llist * using_rels, int id, int scopes_int_p, int record_stats )`

4.29.1.4 `llist* generate_tables_for_scopes_using_relations_noproject ( llist * list_of_lvids, llist * using_rels, int id, int scopes_int_p, int record_stats )`

4.29.1.5 `htable* get_projected_on_variables ( htable * ht, llist * variablelist )`

4.29.1.6 `htable* join_filter_relations_with_all_sols ( llist * htablelist, int nojoin, int nofilter, char * optionalname )`

4.29.1.7 `llist* join_project_multiple_in_multiple_out ( llist * htablelist, llist * list_of_domain_lists, int nofilteroriginal, int id )`

4.29.1.8 `llist* join_project_multiple_in_multiple_out_noextend ( llist * htablelist, llist * list_of_domain_lists, int id )`

4.29.1.9 `llist* join_project_multiple_in_multiple_out_noextend_given_tables ( llist * first_tables, llist * htablelist_ofrest, llist * list_of_domain_lists, int id )`

4.29.1.10 `int process_with_all_sols ( llist * htablelist, active_tuple * ac, options category )`

4.29.1.11 `int process_with_all_sols_first_var ( llist * htablelist, active_tuple * ac, int first_v, options category )`

4.29.1.12 `llist* project_combine_relations ( llist * htables, llist * varlist, int filter )`

4.29.1.13 `htable* project_with_tree ( htable * ht, llist * varlist )`

4.29.1.14 `htable* project_with_tree_payload ( htable * ht, llist * varlist, tree_map ** tree_only, void * get_payloadint param, int param )`

4.29.1.15 `void sort_with_tree ( htable * ht, llist * varlist, int * vmap )`

#### 4.30 hashtable\_search.h File Reference

```
#include "utils.h"
#include "hashtable.h"
#include "llist.h"
#include "active_tuple.h"
```

#### Functions

- `htable * get_projected_on_variables (htable *ht, llist *sepnames)`

- [htable](#) \* [join\\_filter\\_relations\\_with\\_all\\_sols](#) ([llist](#) \*htablelist, int nojoin, int nofliter, char \*optionalname)
- [htable](#) \* [project\\_with\\_tree](#) ([htable](#) \*ht, [llist](#) \*varlist)
- [htable](#) \* [project\\_with\\_tree\\_payload](#) ([htable](#) \*ht, [llist](#) \*varlist, [tree\\_map](#) \*\*tree\_only, void \*get\_payload(int param), int param)
- [llist](#) \* [project\\_combine\\_relations](#) ([llist](#) \*htables, [llist](#) \*variablelist, int filter)
- [llist](#) \* [generate\\_tables\\_for\\_scopes\\_using\\_relations](#) ([llist](#) \*list\_of\_lvids, [llist](#) \*using\_rels, int id, int scopes\_int\_p, int record\_stats)
- [llist](#) \* [generate\\_tables\\_for\\_scopes\\_using\\_relations\\_noproject](#) ([llist](#) \*list\_of\_lvids, [llist](#) \*using\_rels, int id, int scopes\_int\_p, int record\_stats)
- int [filter\\_relations\\_with\\_1\\_sol](#) ([llist](#) \*htablelist)
- int [filter\\_this\\_relation\\_with\\_1\\_sol](#) ([htable](#) \*htfilter, [llist](#) \*given\_htablelist, [set](#) \*undo\_set, [light\\_stack](#) \*undo\_stack)
- [llist](#) \* [join\\_project\\_multiple\\_in\\_multiple\\_out](#) ([llist](#) \*htablelist, [llist](#) \*list\_of\_domain\_lists, int nofilteroriginal, int id)
- void [sort\\_with\\_tree](#) ([htable](#) \*ht, [llist](#) \*varlist, int \*vmap)
- void [find\\_all\\_supports\\_with\\_all\\_solutions\\_with\\_fc\\_noextend](#) ([llist](#) \*htablelist, [llist](#) \*vars)
- [llist](#) \* [join\\_project\\_multiple\\_in\\_multiple\\_out\\_noextend](#) ([llist](#) \*htablelist, [llist](#) \*list\_of\_domain\_lists, int id)
- [llist](#) \* [join\\_project\\_multiple\\_in\\_multiple\\_out\\_noextend\\_given\\_tables](#) ([llist](#) \*first\_tables, [llist](#) \*htablelist\_ofrest, [llist](#) \*list\_of\_domain\_lists, int id)
- int [process\\_with\\_all\\_sols](#) ([llist](#) \*htablelist, [active\\_tuple](#) \*ac, [options](#) category)
- int [process\\_with\\_all\\_sols\\_first\\_var](#) ([llist](#) \*htablelist, [active\\_tuple](#) \*ac, int first\_v, [options](#) category)

### 4.30.1 Detailed Description

Provides tree-based algorithms for processing relations.

### 4.30.2 Function Documentation

4.30.2.1 int [filter\\_relations\\_with\\_1\\_sol](#) ( [llist](#) \* *htablelist* )

4.30.2.2 int [filter\\_this\\_relation\\_with\\_1\\_sol](#) ( [htable](#) \* *htfilter*, [llist](#) \* *given\_htablelist*, [set](#) \* *undo\_set*, [light\\_stack](#) \* *undo\_stack* )

4.30.2.3 void [find\\_all\\_supports\\_with\\_all\\_solutions\\_with\\_fc\\_noextend](#) ( [llist](#) \* *htablelist*, [llist](#) \* *vars* )

4.30.2.4 [llist](#)\* [generate\\_tables\\_for\\_scopes\\_using\\_relations](#) ( [llist](#) \* *list\_of\_lvids*, [llist](#) \* *using\_rels*, int *id*, int *scopes\_int\_p*, int *record\_stats* )

4.30.2.5 [llist](#)\* [generate\\_tables\\_for\\_scopes\\_using\\_relations\\_noproject](#) ( [llist](#) \* *list\_of\_lvids*, [llist](#) \* *using\_rels*, int *id*, int *scopes\_int\_p*, int *record\_stats* )

4.30.2.6 [htable](#)\* [get\\_projected\\_on\\_variables](#) ( [htable](#) \* *ht*, [llist](#) \* *sepnames* )

4.30.2.7 [htable](#)\* [join\\_filter\\_relations\\_with\\_all\\_sols](#) ( [llist](#) \* *htablelist*, int *nojoin*, int *nofliter*, char \* *optionalname* )

4.30.2.8 [llist](#)\* [join\\_project\\_multiple\\_in\\_multiple\\_out](#) ( [llist](#) \* *htablelist*, [llist](#) \* *list\_of\_domain\_lists*, int *nofilteroriginal*, int *id* )

4.30.2.9 [llist](#)\* [join\\_project\\_multiple\\_in\\_multiple\\_out\\_noextend](#) ( [llist](#) \* *htablelist*, [llist](#) \* *list\_of\_domain\_lists*, int *id* )

4.30.2.10 [llist](#)\* [join\\_project\\_multiple\\_in\\_multiple\\_out\\_noextend\\_given\\_tables](#) ( [llist](#) \* *first\_tables*, [llist](#) \* *htablelist\_ofrest*, [llist](#) \* *list\_of\_domain\_lists*, int *id* )

4.30.2.11 int [process\\_with\\_all\\_sols](#) ( [llist](#) \* *htablelist*, [active\\_tuple](#) \* *ac*, [options](#) *category* )

- 4.30.2.12 `int process_with_all_sols_first_var ( llist *htablelist, active_tuple *ac, int first_v, options category )`
- 4.30.2.13 `llist* project_combine_relations ( llist *htables, llist *variablelist, int filter )`
- 4.30.2.14 `htable* project_with_tree ( htable *ht, llist *varlist )`
- 4.30.2.15 `htable* project_with_tree_payload ( htable *ht, llist *varlist, tree_map **tree_only, void *get_payloadint param, int param )`
- 4.30.2.16 `void sort_with_tree ( htable *ht, llist *varlist, int *vmap )`

## 4.31 heap.c File Reference

```
#include <stdio.h>
#include "heap.h"
#include "utils.h"
```

### Functions

- `heap * new_heap` (int size)
- `void destroy_heap` (heap \*h)
- `int heap_parent` (int i)
- `int heap_left` (int i)
- `int heap_right` (int i)
- `void heap_exchange` (int i, int j, heap \*h, void set\_heap\_position(int, void \*))
- `void min_heapify` (heap \*h, int i, int get\_heap\_value(void \*), void set\_heap\_position(int, void \*))
- `void build_min_heap` (heap \*h, int get\_heap\_value(void \*), void set\_heap\_position(int, void \*))
- `void * heap_min` (heap \*h)
- `void * heap_extract_min` (heap \*h, int get\_heap\_value(void \*), void set\_heap\_position(int, void \*))
- `void heap_move_up` (heap \*h, int i, int get\_heap\_value(void \*), void set\_heap\_position(int, void \*))
- `void heap_update` (heap \*h, int i, int val, int get\_heap\_value(void \*), void set\_heap\_value(int, void \*), void set\_heap\_position(int, void \*))
- `void heap_insert` (heap \*h, void \*item, int get\_heap\_value(void \*), void set\_heap\_value(int, void \*), void set\_heap\_position(int, void \*))
- `void heap_delete` (heap \*h, int i, int get\_heap\_value(void \*), void set\_heap\_value(int, void \*), void set\_heap\_position(int, void \*))
- `void print_heap` (heap \*h, int get\_heap\_value(void \*))

### 4.31.1 Function Documentation

- 4.31.1.1 `void build_min_heap ( heap * h, int get_heap_valuevoid *, void set_heap_positionint, void * )`
- 4.31.1.2 `void destroy_heap ( heap * h )`
- 4.31.1.3 `void heap_delete ( heap * h, int i, int get_heap_valuevoid *, void set_heap_valueint, void *, void set_heap_positionint, void * )`
- 4.31.1.4 `void heap_exchange ( int i, int j, heap * h, void set_heap_positionint, void * )`
- 4.31.1.5 `void* heap_extract_min ( heap * h, int get_heap_valuevoid *, void set_heap_positionint, void * )`
- 4.31.1.6 `void heap_insert ( heap * h, void * item, int get_heap_valuevoid *, void set_heap_valueint, void *, void set_heap_positionint, void * )`



- 4.31.1.7 `int heap_left ( int i )`
- 4.31.1.8 `void* heap_min ( heap * h )`
- 4.31.1.9 `void heap_move_up ( heap * h, int i, int get_heap_valuevoid *, void set_heap_positionint, void * )`
- 4.31.1.10 `int heap_parent ( int i )`
- 4.31.1.11 `int heap_right ( int i )`
- 4.31.1.12 `void heap_update ( heap * h, int i, int val, int get_heap_valuevoid *, void set_heap_valueint, void *, void set_heap_positionint, void * )`
- 4.31.1.13 `void min_heapify ( heap * h, int i, int get_heap_valuevoid *, void set_heap_positionint, void * )`
- 4.31.1.14 `heap* new_heap ( int size )`
- 4.31.1.15 `void print_heap ( heap * h, int get_heap_valuevoid * )`

## 4.32 heap.h File Reference

### Classes

- struct [heap](#)

### Macros

- #define [INF](#) 2147483646

### Typedefs

- typedef struct [heap](#) [heap](#)

### Functions

- [heap \\* new\\_heap](#) (int size)
- void [destroy\\_heap](#) ([heap \\*h](#))
- void [heap\\_update](#) ([heap \\*h](#), int i, int val, int [get\\_heap\\_value](#)(void \*), void [set\\_heap\\_value](#)(int, void \*), void [set\\_heap\\_position](#)(int, void \*))
- int [heap\\_parent](#) (int i)
- int [heap\\_left](#) (int i)
- int [heap\\_right](#) (int i)
- void [heap\\_exchange](#) (int i, int j, [heap \\*h](#), void [set\\_heap\\_position](#)(int, void \*))
- void [min\\_heapify](#) ([heap \\*h](#), int i, int [get\\_heap\\_value](#)(void \*), void [set\\_heap\\_position](#)(int, void \*))
- void [build\\_min\\_heap](#) ([heap \\*h](#), int [get\\_heap\\_value](#)(void \*), void [set\\_heap\\_position](#)(int, void \*))
- void \* [heap\\_min](#) ([heap \\*h](#))
- void \* [heap\\_extract\\_min](#) ([heap \\*h](#), int [get\\_heap\\_value](#)(void \*), void [set\\_heap\\_position](#)(int, void \*))
- void [heap\\_move\\_up](#) ([heap \\*h](#), int i, int [get\\_heap\\_value](#)(void \*), void [set\\_heap\\_position](#)(int, void \*))
- void [heap\\_insert](#) ([heap \\*h](#), void \*item, int [get\\_heap\\_value](#)(void \*), void [set\\_heap\\_value](#)(int, void \*), void [set\\_heap\\_position](#)(int, void \*))
- void [heap\\_delete](#) ([heap \\*h](#), int i, int [get\\_heap\\_value](#)(void \*), void [set\\_heap\\_value](#)(int, void \*), void [set\\_heap\\_position](#)(int, void \*))
- void [print\\_heap](#) ([heap \\*hd](#), int [get\\_heap\\_value](#)(void \*))

### 4.32.1 Detailed Description

Provides a heap implementation.

### 4.32.2 Macro Definition Documentation

4.32.2.1 `#define INF 2147483646`

### 4.32.3 Typedef Documentation

4.32.3.1 `typedef struct heap heap`

### 4.32.4 Function Documentation

4.32.4.1 `void build_min_heap ( heap * h, int get_heap_valuevoid *, void set_heap_positionint, void * )`

4.32.4.2 `void destroy_heap ( heap * h )`

4.32.4.3 `void heap_delete ( heap * h, int i, int get_heap_valuevoid *, void set_heap_valueint, void *, void set_heap_positionint, void * )`

4.32.4.4 `void heap_exchange ( int i, int j, heap * h, void set_heap_positionint, void * )`

4.32.4.5 `void* heap_extract_min ( heap * h, int get_heap_valuevoid *, void set_heap_positionint, void * )`

4.32.4.6 `void heap_insert ( heap * h, void * item, int get_heap_valuevoid *, void set_heap_valueint, void *, void set_heap_positionint, void * )`

4.32.4.7 `int heap_left ( int i )`

4.32.4.8 `void* heap_min ( heap * h )`

4.32.4.9 `void heap_move_up ( heap * h, int i, int get_heap_valuevoid *, void set_heap_positionint, void * )`

4.32.4.10 `int heap_parent ( int i )`

4.32.4.11 `int heap_right ( int i )`

4.32.4.12 `void heap_update ( heap * h, int i, int val, int get_heap_valuevoid *, void set_heap_valueint, void *, void set_heap_positionint, void * )`

4.32.4.13 `void min_heapify ( heap * h, int i, int get_heap_valuevoid *, void set_heap_positionint, void * )`

4.32.4.14 `heap* new_heap ( int size )`

4.32.4.15 `void print_heap ( heap * hd, int get_heap_valuevoid * )`

## 4.33 histogram.c File Reference

```
#include "histogram.h"
#include "utils.h"
```

## 4.34 histogram.h File Reference

### 4.34.1 Detailed Description

Provides histogram implementation for the relations.

## 4.35 llist.c File Reference

```
#include "llist.h"
#include "domains.h"
#include "utils.h"
```

### Functions

- [llist\\_node \\* new\\_lnode](#) (void)
- [llist \\* new\\_llist](#) (void)
- void [destroy\\_list\\_nodes\\_only](#) (llist \*list)
- void [destroy\\_list\\_nodes](#) (llist\_node \*nodes)
- void [destroy\\_list\\_nodes\\_body](#) (llist\_node \*nodes)
- void [destroy\\_list](#) (llist \*list)
- void [destroy\\_list\\_body](#) (llist \*list)
- void [add\\_int](#) (int ibody, llist \*list)
- void [add\\_node](#) (void \*body, llist \*list)
- void [add\\_node\\_tail](#) (void \*body, llist \*list)
- void [add\\_node\\_tail\\_distinct](#) (void \*body, llist \*list)
- void [add\\_int\\_tail\\_distinct](#) (int ibody, llist \*list)
- void [add\\_int\\_tail](#) (int ibody, llist \*list)
- void [add\\_this\\_node\\_tail](#) (llist\_node \*node, llist \*list)
- void [add\\_this\\_node](#) (llist\_node \*node, llist \*list)
- [llist\\_node \\* find\\_remove\\_node\\_of\\_body](#) (void \*value, int compare(void \*value, void \*bodyb), llist \*list)
- int [restore\\_element\\_to\\_same\\_position](#) (llist\_node \*node, llist \*list)
- [llist\\_node \\* remove\\_node\\_no\\_check\\_no\\_null](#) (llist\_node \*node, llist \*list)
- [llist\\_node \\* remove\\_node\\_no\\_check](#) (llist\_node \*node, llist \*list)
- void [move\\_to\\_tail](#) (llist\_node \*node, llist \*list)
- void [move\\_to\\_head](#) (llist\_node \*node, llist \*list)
- [llist\\_node \\* remove\\_node](#) (llist\_node \*node, llist \*list)
- void [remove\\_node\\_of\\_body](#) (void \*body, llist \*list)
- [llist\\_node \\* remove\\_single\\_node\\_of\\_body](#) (void \*body, llist \*list)
- [llist\\_node \\* find\\_node\\_of\\_value](#) (void \*value, int compare(void \*value, void \*bodyb), llist \*list)
- [llist\\_node \\* find\\_node\\_of\\_ivalue](#) (int value, llist \*list)
- [llist\\_node \\* get\\_node\\_of\\_body](#) (void \*body, llist \*list)
- [llist\\_node \\* remove\\_head](#) (llist \*list)
- [llist\\_node \\* remove\\_tail](#) (llist \*list)
- void [move\\_llist\\_to](#) (llist \*list, llist \*destination\_list)
- void [copy\\_llist\\_to](#) (llist \*list, llist \*destination\_list)
- void [copy\\_llist\\_to\\_distinct](#) (llist \*list, llist \*destination\_list)
- [llist \\* copy\\_int\\_llist\\_add\\_pos](#) (llist \*list, int pos0)
- [llist \\* copy\\_llist](#) (llist \*list)
- [llist \\* copy\\_value\\_llist](#) (llist \*list)
- int [is\\_empty](#) (llist \*list)
- int [max\\_int](#) (llist \*list, int get\_value(void \*body))
- int [min\\_int](#) (llist \*list, int get\_value(void \*body))

- `llist_node * remove_max (llist *list, int get_value(void *body))`
- `llist_node * remove_min (llist *list, int get_value(void *body))`
- `llist_node * get_max (llist *list, int get_value(void *body))`
- `llist_node * get_min (llist *list, int get_value(void *body))`
- `llist_node * remove_node_of_ibody (int ibody, llist *list)`
- `llist_node * get_node_of_ibody (int ibody, llist *list)`
- `light_stack * new_light_stack (void)`
- `light_stack_node * new_light_stack_node (int value)`
- `void push_to_light_stack (int value, void *body, light_stack *lis)`
- `int pop_light_stack (light_stack *lis, void **body_holder)`
- `void push_s (void *item, llist *stack)`
- `void * pop_s (llist *stack)`
- `void * peek_s (llist *stack)`
- `void * llist_dequeue (llist *queue)`
- `void llist_enqueue (void *object, llist *queue)`

### 4.35.1 Function Documentation

- 4.35.1.1 `void add_int ( int ibody, llist * list )`
- 4.35.1.2 `void add_int_tail ( int ibody, llist * list )`
- 4.35.1.3 `void add_int_tail_distinct ( int ibody, llist * list )`
- 4.35.1.4 `void add_node ( void * body, llist * list )`
- 4.35.1.5 `void add_node_tail ( void * body, llist * list )`
- 4.35.1.6 `void add_node_tail_distinct ( void * body, llist * list )`
- 4.35.1.7 `void add_this_node ( llist_node * node, llist * list )`
- 4.35.1.8 `void add_this_node_tail ( llist_node * node, llist * list )`
- 4.35.1.9 `llist* copy_int_llist_add_pos ( llist * list, int pos0 )`
- 4.35.1.10 `llist* copy_llist ( llist * list )`
- 4.35.1.11 `void copy_llist_to ( llist * list, llist * destination_list )`
- 4.35.1.12 `void copy_llist_to_distinct ( llist * list, llist * destination_list )`
- 4.35.1.13 `llist* copy_value_llist ( llist * list )`
- 4.35.1.14 `void destroy_list ( llist * list )`
- 4.35.1.15 `void destroy_list_body ( llist * list )`
- 4.35.1.16 `void destroy_list_nodes ( llist_node * nodes )`
- 4.35.1.17 `void destroy_list_nodes_body ( llist_node * nodes )`
- 4.35.1.18 `void destroy_list_nodes_only ( llist * list )`
- 4.35.1.19 `llist_node* find_node_of_ivalue ( int value, llist * list )`

- 4.35.1.20 `llist_node* find_node_of_value ( void * value, int comparevoid *value, void *bodyb, llist * list )`
- 4.35.1.21 `llist_node* find_remove_node_of_body ( void * value, int comparevoid *value, void *bodyb, llist * list )`
- 4.35.1.22 `llist_node* get_max ( llist * list, int get_valuevoid *body )`
- 4.35.1.23 `llist_node* get_min ( llist * list, int get_valuevoid *body )`
- 4.35.1.24 `llist_node* get_node_of_body ( void * body, llist * list )`
- 4.35.1.25 `llist_node* get_node_of_ibody ( int ibody, llist * list )`
- 4.35.1.26 `int is_empty ( llist * list )`
- 4.35.1.27 `void* llist_dequeue ( llist * queue )`
- 4.35.1.28 `void llist_enqueue ( void * object, llist * queue )`
- 4.35.1.29 `int max_int ( llist * list, int get_valuevoid *body )`
- 4.35.1.30 `int min_int ( llist * list, int get_valuevoid *body )`
- 4.35.1.31 `void move_llist_to ( llist * list, llist * destination_list )`
- 4.35.1.32 `void move_to_head ( llist_node * node, llist * list )`
- 4.35.1.33 `void move_to_tail ( llist_node * node, llist * list )`
- 4.35.1.34 `light_stack* new_light_stack ( void )`
- 4.35.1.35 `light_stack_node* new_light_stack_node ( int value )`
- 4.35.1.36 `llist* new_llist ( void )`
- 4.35.1.37 `llist_node* new_llnode ( void )`
- 4.35.1.38 `void* peek_s ( llist * stack )`
- 4.35.1.39 `int pop_light_stack ( light_stack * lis, void ** body_holder )`
- 4.35.1.40 `void* pop_s ( llist * stack )`
- 4.35.1.41 `void push_s ( void * item, llist * stack )`
- 4.35.1.42 `void push_to_light_stack ( int value, void * body, light_stack * lis )`
- 4.35.1.43 `llist_node* remove_head ( llist * list )`
- 4.35.1.44 `llist_node* remove_max ( llist * list, int get_valuevoid *body )`
- 4.35.1.45 `llist_node* remove_min ( llist * list, int get_valuevoid *body )`
- 4.35.1.46 `llist_node* remove_node ( llist_node * node, llist * list )`
- 4.35.1.47 `llist_node* remove_node_no_check ( llist_node * node, llist * list )`

4.35.1.48 `llist_node*` `remove_node_no_check_no_null ( llist_node * node, llist * list )`

4.35.1.49 `void` `remove_node_of_body ( void * body, llist * list )`

4.35.1.50 `llist_node*` `remove_node_of_ibody ( int ibody, llist * list )`

4.35.1.51 `llist_node*` `remove_single_node_of_body ( void * body, llist * list )`

4.35.1.52 `llist_node*` `remove_tail ( llist * list )`

4.35.1.53 `int` `restore_element_to_same_position ( llist_node * node, llist * list )`

## 4.36 llist.h File Reference

```
#include <stdlib.h>
```

### Classes

- struct [llist\\_node](#)
- struct [llist](#)
- struct [light\\_stack](#)
- struct [light\\_stack\\_node](#)

### Typedefs

- typedef struct [llist\\_node](#) [llist\\_node](#)
- typedef struct [llist](#) [llist](#)
- typedef struct [light\\_stack](#) [light\\_stack](#)
- typedef struct [light\\_stack\\_node](#) [light\\_stack\\_node](#)

### Functions

- `llist_node *` `new_lnode` (void)
- `llist *` `new_llist` (void)
- void `add_int` (int ibody, `llist *`list)
- void `add_node` (void \*body, `llist *`list)
- void `add_node_tail` (void \*body, `llist *`list)
- void `add_node_tail_distinct` (void \*body, `llist *`list)
- void `add_int_tail` (int ibody, `llist *`list)
- void `add_int_tail_distinct` (int ibody, `llist *`list)
- void `add_this_node_tail` (`llist_node *`node, `llist *`list)
- void `add_this_node` (`llist_node *`node, `llist *`list)
- `llist_node *` `remove_node` (`llist_node *`node, `llist *`list)
- int `restore_element_to_same_position` (`llist_node *`node, `llist *`list)
- `llist_node *` `remove_node_no_check_no_null` (`llist_node *`node, `llist *`list)
- `llist_node *` `remove_node_no_check` (`llist_node *`node, `llist *`list)
- void `remove_node_of_body` (void \*body, `llist *`list)
- `llist_node *` `find_remove_node_of_body` (void \*value, int compare(void \*value, void \*bodyb), `llist *`list)
- `llist_node *` `get_node_of_body` (void \*body, `llist *`list)
- `llist_node *` `find_node_of_value` (void \*value, int compare(void \*value, void \*bodyb), `llist *`list)
- `llist_node *` `find_node_of_ivalue` (int value, `llist *`list)
- `llist_node *` `remove_single_node_of_body` (void \*body, `llist *`list)

- `llist_node * remove_head (llist *list)`
- `llist_node * remove_tail (llist *list)`
- `llist * copy_value_llist (llist *list)`
- `llist * copy_llist (llist *list)`
- `void copy_llist_to (llist *list, llist *destination_list)`
- `void copy_llist_to_distinct (llist *list, llist *destination_list)`
- `llist * copy_int_llist_add_pos (llist *list, int pos0)`
- `int is_empty (llist *list)`
- `void destroy_list_nodes_only (llist *list)`
- `void destroy_list_nodes (llist_node *list_node)`
- `void destroy_list_nodes_body (llist_node *list_node)`
- `void destroy_list (llist *list)`
- `void destroy_list_body (llist *list)`
- `int max_int (llist *list, int get_value(void *body))`
- `llist_node * remove_max (llist *list, int get_value(void *body))`
- `llist_node * remove_min (llist *list, int get_value(void *body))`
- `llist_node * get_max (llist *list, int get_value(void *body))`
- `llist_node * get_min (llist *list, int get_value(void *body))`
- `void move_to_tail (llist_node *node, llist *list)`
- `void move_to_head (llist_node *node, llist *list)`
- `llist_node * remove_node_of_ibody (int ibody, llist *list)`
- `llist_node * get_node_of_ibody (int ibody, llist *list)`
- `int min_int (llist *list, int get_value(void *body))`
- `void move_llist_to (llist *list, llist *destination_list)`
- `light_stack * new_light_stack (void)`
- `light_stack_node * new_light_stack_node (int value)`
- `void push_to_light_stack (int value, void *body, light_stack *lis)`
- `int pop_light_stack (light_stack *lis, void **body_holder)`
- `void push_s (void *item, llist *stack)`
- `void * pop_s (llist *stack)`
- `void * peek_s (llist *stack)`
- `void * llist_dequeue (llist *queue)`
- `void llist_enqueue (void *object, llist *queue)`

#### 4.36.1 Detailed Description

Provides linked-list implementation.

#### 4.36.2 Typedef Documentation

4.36.2.1 `typedef struct light_stack light_stack`

4.36.2.2 `typedef struct light_stack_node light_stack_node`

4.36.2.3 `typedef struct llist llist`

4.36.2.4 `typedef struct llist_node llist_node`

#### 4.36.3 Function Documentation

4.36.3.1 `void add_int ( int ibody, llist * list )`

4.36.3.2 `void add_int_tail ( int ibody, llist * list )`

- 4.36.3.3 void add\_int\_tail\_distinct ( int *ibody*, llist \* *list* )
- 4.36.3.4 void add\_node ( void \* *body*, llist \* *list* )
- 4.36.3.5 void add\_node\_tail ( void \* *body*, llist \* *list* )
- 4.36.3.6 void add\_node\_tail\_distinct ( void \* *body*, llist \* *list* )
- 4.36.3.7 void add\_this\_node ( llist\_node \* *node*, llist \* *list* )
- 4.36.3.8 void add\_this\_node\_tail ( llist\_node \* *node*, llist \* *list* )
- 4.36.3.9 llist\* copy\_int\_llist\_add\_pos ( llist \* *list*, int *pos0* )
- 4.36.3.10 llist\* copy\_llist ( llist \* *list* )
- 4.36.3.11 void copy\_llist\_to ( llist \* *list*, llist \* *destination\_list* )
- 4.36.3.12 void copy\_llist\_to\_distinct ( llist \* *list*, llist \* *destination\_list* )
- 4.36.3.13 llist\* copy\_value\_llist ( llist \* *list* )
- 4.36.3.14 void destroy\_list ( llist \* *list* )
- 4.36.3.15 void destroy\_list\_body ( llist \* *list* )
- 4.36.3.16 void destroy\_list\_nodes ( llist\_node \* *llist\_node* )
- 4.36.3.17 void destroy\_list\_nodes\_body ( llist\_node \* *llist\_node* )
- 4.36.3.18 void destroy\_list\_nodes\_only ( llist \* *list* )
- 4.36.3.19 llist\_node\* find\_node\_of\_ivalue ( int *value*, llist \* *list* )
- 4.36.3.20 llist\_node\* find\_node\_of\_value ( void \* *value*, int *comparevoid \*value*, void \**bodyb*, llist \* *list* )
- 4.36.3.21 llist\_node\* find\_remove\_node\_of\_body ( void \* *value*, int *comparevoid \*value*, void \**bodyb*, llist \* *list* )
- 4.36.3.22 llist\_node\* get\_max ( llist \* *list*, int *get\_valuevoid \*body* )
- 4.36.3.23 llist\_node\* get\_min ( llist \* *list*, int *get\_valuevoid \*body* )
- 4.36.3.24 llist\_node\* get\_node\_of\_body ( void \* *body*, llist \* *list* )
- 4.36.3.25 llist\_node\* get\_node\_of\_ibody ( int *ibody*, llist \* *list* )
- 4.36.3.26 int is\_empty ( llist \* *list* )
- 4.36.3.27 void\* llist\_dequeue ( llist \* *queue* )
- 4.36.3.28 void llist\_enqueue ( void \* *object*, llist \* *queue* )
- 4.36.3.29 int max\_int ( llist \* *list*, int *get\_valuevoid \*body* )
- 4.36.3.30 int min\_int ( llist \* *list*, int *get\_valuevoid \*body* )



- 4.36.3.31 void move\_llist\_to ( llist \* list, llist \* destination\_list )
- 4.36.3.32 void move\_to\_head ( llist\_node \* node, llist \* list )
- 4.36.3.33 void move\_to\_tail ( llist\_node \* node, llist \* list )
- 4.36.3.34 light\_stack\* new\_light\_stack ( void )
- 4.36.3.35 light\_stack\_node\* new\_light\_stack\_node ( int value )
- 4.36.3.36 llist\* new\_llist ( void )
- 4.36.3.37 llist\_node\* new\_llnode ( void )
- 4.36.3.38 void\* peek\_s ( llist \* stack )
- 4.36.3.39 int pop\_light\_stack ( light\_stack \* lis, void \*\* body\_holder )
- 4.36.3.40 void\* pop\_s ( llist \* stack )
- 4.36.3.41 void push\_s ( void \* item, llist \* stack )
- 4.36.3.42 void push\_to\_light\_stack ( int value, void \* body, light\_stack \* lis )
- 4.36.3.43 llist\_node\* remove\_head ( llist \* list )
- 4.36.3.44 llist\_node\* remove\_max ( llist \* list, int get\_valuevoid \*body )
- 4.36.3.45 llist\_node\* remove\_min ( llist \* list, int get\_valuevoid \*body )
- 4.36.3.46 llist\_node\* remove\_node ( llist\_node \* node, llist \* list )
- 4.36.3.47 llist\_node\* remove\_node\_no\_check ( llist\_node \* node, llist \* list )
- 4.36.3.48 llist\_node\* remove\_node\_no\_check\_no\_null ( llist\_node \* node, llist \* list )
- 4.36.3.49 void remove\_node\_of\_body ( void \* body, llist \* list )
- 4.36.3.50 llist\_node\* remove\_node\_of\_ibody ( int ibody, llist \* list )
- 4.36.3.51 llist\_node\* remove\_single\_node\_of\_body ( void \* body, llist \* list )
- 4.36.3.52 llist\_node\* remove\_tail ( llist \* list )
- 4.36.3.53 int restore\_element\_to\_same\_position ( llist\_node \* node, llist \* list )

## 4.37 main.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <string.h>
#include <math.h>
#include <getopt.h>
#include "parser/C_CSPParserCallback.h"
#include "globals.h"
#include "procedures.h"
#include "utils.h"
#include "ac.h"
#include "randomize.h"
#include "query_graph.h"
#include "sac.h"
#include "rpwc.h"
#include "terminal.h"
#include "do_db.h"
#include "parsers.h"
#include "normalize_constraints.h"
#include "decomposition.h"
#include "oneall.h"
#include "solution.h"
#include <signal.h>
```

### Functions

- void [sigxcpu\\_handler](#) (int signo)
- int [main](#) (int argc, char \*\*argv)
- void [check\\_ok2](#) (variables \*vars)
- void [check\\_ok](#) (main\_structure \*m\_s)

### Variables

- [main\\_structure \\* m\\_s\\_g](#)

#### 4.37.1 Detailed Description

Provides the code for parsing the input parameters and runs the program.

Finding One solution:

(for pre-processing only, remove '-Sx')

```
GAC :: -f [file] -k -uDD -DGA -F -Sx -v-1
maxRPWC :: -f [file] -k -uDD -DRP -F -Sx -v-1
```

```
Random: cl+proj-wR(*,2)C :: -f [file] -k -Tmfrstonenosp -xpb -qms -Drr -s2 -b2s -uDD -F -Sx
Random: cl+proj-wR(*,3)C :: -f [file] -k -Tmfrstonenosp -xpb -qms -Drr -s3 -b2s -uDD -F -Sx
Random: cl+proj-wR(*,4)C :: -f [file] -k -Tmfrstonenosp -xpb -qms -Drr -s4 -b2s -uDD -F -Sx
Random: cl+proj-wR(*,|C)C :: -f [file] -k -Tmfrstonenosp -xpb -qms -s0 -b2s -uDD -F -Sx
```

Random: cl-wR(\*,2)C :: -f [file] -k -Tmfrstonenosp -xss -qms -Drr -s2 -b2s -uDD -F -Sx  
 Random: cl-wR(\*,3)C :: -f [file] -k -Tmfrstonenosp -xss -qms -Drr -s3 -b2s -uDD -F -Sx  
 Random: cl-wR(\*,4)C :: -f [file] -k -Tmfrstonenosp -xss -qms -Drr -s4 -b2s -uDD -F -Sx  
 Random: cl-wR(\*,|C|)C :: -f [file] -k -Tmfrstonenosp -xss -qms -s0 -b2s -uDD -F -Sx

wR(\*,2)C :: -f [file] -k -Tmfrstonenosp -xpb -Drr -s2 -b1 -uDD -F -Sx -v-1  
 wR(\*,3)C :: -f [file] -k -Tmfrstonenosp -xpb -Drr -s3 -b1 -uDD -F -Sx -v-1  
 wR(\*,4)C :: -f [file] -k -Tmfrstonenosp -xpb -Drr -s4 -b1 -uDD -F -Sx -v-1

Static: cl-wR(\*,2)C :: -f [file] -k -Tmfrstonenosp -xss -qmr -Drr -s2 -b2n0 -uDD -F -Sx  
 Static: cl-wR(\*,3)C :: -f [file] -k -Tmfrstonenosp -xss -qmr -Drr -s3 -b2n0 -uDD -F -Sx  
 Static: cl-wR(\*,4)C :: -f [file] -k -Tmfrstonenosp -xss -qmr -Drr -s4 -b2n0 -uDD -F -Sx  
 Static: cl-wR(\*,|C|)C :: -f [file] -k -Tmfrstonenosp -xss -qmr -s0 -b2n0 -uDD -F -Sx

Static: cl+proj-wR(\*,2)C :: -f [file] -k -Tmfrstonenosp -xpb -qmr -Drr -s2 -b2n0 -uDD -F -Sx  
 Static: cl+proj-wR(\*,3)C :: -f [file] -k -Tmfrstonenosp -xpb -qmr -Drr -s3 -b2n0 -uDD -F -Sx  
 Static: cl+proj-wR(\*,4)C :: -f [file] -k -Tmfrstonenosp -xpb -qmr -Drr -s4 -b2n0 -uDD -F -Sx  
 Static: cl+proj-wR(\*,|C|)C :: -f [file] -k -Tmfrstonenosp -xpb -qmr -s0 -b2n0 -uDD -F -Sx

Static: cl+bin-wR(\*,2)C :: -f [file] -k -Tmfrstonebinsp+ -xpb -qmr -Drr -s2 -b2n0 -uDD -F -Sx  
 Static: cl+bin-wR(\*,3)C :: -f [file] -k -Tmfrstonebinsp+ -xpb -qmr -Drr -s3 -b2n0 -uDD -F -Sx  
 Static: cl+bin-wR(\*,4)C :: -f [file] -k -Tmfrstonebinsp+ -xpb -qmr -Drr -s4 -b2n0 -uDD -F -Sx  
 Static: cl+bin-wR(\*,|C|)C :: -f [file] -k -Tmfrstonebinsp+ -xpb -qmr -s0 -b2n0 -uDD -F -Sx

Static: cl+clq-wR(\*,2)C :: -f [file] -k -Tmfrstonemaxcsp+ -xpb -qmr -Drr -s2 -b2n0 -uDD -F -Sx  
 Static: cl+clq-wR(\*,3)C :: -f [file] -k -Tmfrstonemaxcsp+ -xpb -qmr -Drr -s3 -b2n0 -uDD -F -Sx  
 Static: cl+clq-wR(\*,4)C :: -f [file] -k -Tmfrstonemaxcsp+ -xpb -qmr -Drr -s4 -b2n0 -uDD -F -Sx  
 Static: cl+clq-wR(\*,|C|)C :: -f [file] -k -Tmfrstonemaxcsp+ -xpb -qmr -s0 -b2n0 -uDD -F -Sx

Priority: cl+proj-wR(\*,2)C :: -f [file] -k -Tmfrstonenosp -xss -qm12 -Drr -s2 -b2n2 -uDD -F -Sx':  
 Priority: cl+proj-wR(\*,3)C :: -f [file] -k -Tmfrstonenosp -xss -qm12 -Drr -s3 -b2n2 -uDD -F -Sx':  
 Priority: cl+proj-wR(\*,4)C :: -f [file] -k -Tmfrstonenosp -xss -qm12 -Drr -s4 -b2n2 -uDD -F -Sx':  
 Priority: cl-wR(\*,|C|)C :: -f [file] -k -Tmfrstonenosp -xss -qm12 -s0 -b2n2 -uDD -F -Sx

Priority: cl+proj-wR(\*,2)C :: -f [file] -k -Tmfrstonenosp -xpb -qm12 -Drr -s2 -b2n2 -uDD -F -Sx  
 Priority: cl+proj-wR(\*,3)C :: -f [file] -k -Tmfrstonenosp -xpb -qm12 -Drr -s3 -b2n2 -uDD -F -Sx  
 Priority: cl+proj-wR(\*,4)C :: -f [file] -k -Tmfrstonenosp -xpb -qm12 -Drr -s4 -b2n2 -uDD -F -Sx  
 Priority: cl+proj-wR(\*,|C|)C :: -f [file] -k -Tmfrstonenosp -xpb -qm12 -s0 -b2n2 -uDD -F -Sx

Dynamic: cl-wR(\*,2)C :: -f [file] -f -k -Tmfrstonenosp -xss -qm122 -Drr -s2 -b2n2 -uDD -F -Sx':  
 Dynamic: cl-wR(\*,3)C :: -f [file] -f -k -Tmfrstonenosp -xss -qm122 -Drr -s3 -b2n2 -uDD -F -Sx':  
 Dynamic: cl-wR(\*,4)C :: -f [file] -f -k -Tmfrstonenosp -xss -qm122 -Drr -s4 -b2n2 -uDD -F -Sx':  
 Dynamic: cl-wR(\*,|C|)C :: -f [file] -k -Tmfrstonenosp -xss -qm122 -s0 -b2n2 -uDD -F -Sx

Dynamic: cl+proj-wR(\*,2)C :: -f [file] -k -Tmfrstonenosp -xpb -qm122 -Drr -s2 -b2n2 -uDD -F -Sx  
 Dynamic: cl+proj-wR(\*,3)C :: -f [file] -k -Tmfrstonenosp -xpb -qm122 -Drr -s3 -b2n2 -uDD -F -Sx  
 Dynamic: cl+proj-wR(\*,4)C :: -f [file] -k -Tmfrstonenosp -xpb -qm122 -Drr -s4 -b2n2 -uDD -F -Sx  
 Dynamic: cl+proj-wR(\*,|C|)C :: -f [file] -f -k -Tmfrstonenosp -xpb -qm122 -s0 -b2n2 -uDD -F -Sx

## Minimal Network of Each Cluster

```
PerTuple :: -f [file] -k -F -v-10 -Tmfrstonenosp -s0 -xpb -qmr -b2x0
AllSol :: -f [file] -k -F -v-10 -Tmfrstallnosp -s0 -xpb -qmr -b2x0
Features :: -f [file] -k -F -v-10 -Tmfrstnosp -s0 -xpb -qmr -b2x0
```

## Minimal Network of the problem (considered as a single clique)

```
PerTuple :: if [file] -k -F -v-10 -Tmfrstonenosp -s0 -xpb -b1x0 -Sx
AllSol :: if [file] -k -F -v-10 -Tmfrstallnosp -s0 -xpb -b1x0 -Sx
```

## Finding All solutions BTD:

```
GAC :: -f [file] -k -Ubtdd -uDD -DGA -F -Sall -v-1
```

```
wR(*,2)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xpb -Drr -s2 -b1 -uDD -F -Sall -v-1
wR(*,3)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xpb -Drr -s3 -b1 -uDD -F -Sall -v-1
wR(*,4)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xpb -Drr -s4 -b1 -uDD -F -Sall -v-1
```

```
Static: cl-wR(*,2)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xss -qmr -Drr -s2 -b2n0 -uDD -F -Sall
Static: cl-wR(*,3)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xss -qmr -Drr -s3 -b2n0 -uDD -F -Sall
Static: cl-wR(*,4)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xss -qmr -Drr -s4 -b2n0 -uDD -F -Sall
Static: cl-wR(*,|C)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xss -qmr -s0 -b2n0 -uDD -F -Sall
```

```
Static: cl+proj-wR(*,2)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xpb -qmr -Drr -s2 -b2n0 -uDD -F -Sall
Static: cl+proj-wR(*,3)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xpb -qmr -Drr -s3 -b2n0 -uDD -F -Sall
Static: cl+proj-wR(*,4)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xpb -qmr -Drr -s4 -b2n0 -uDD -F -Sall
Static: cl+proj-wR(*,|C)C :: -f [file] -Ubtdd -k -Tmfrstonenosp -xpb -qmr -s0 -b2n0 -uDD -F -Sall
```

```
Static: cl+bin-wR(*,2)C :: -f [file] -Ubtdd -k -Tmfrstonebinsp+ -xpb -qmr -Drr -s2 -b2n0 -uDD -F -Sall
Static: cl+bin-wR(*,3)C :: -f [file] -Ubtdd -k -Tmfrstonebinsp+ -xpb -qmr -Drr -s3 -b2n0 -uDD -F -Sall
Static: cl+bin-wR(*,4)C :: -f [file] -Ubtdd -k -Tmfrstonebinsp+ -xpb -qmr -Drr -s4 -b2n0 -uDD -F -Sall
Static: cl+bin-wR(*,|C)C :: -f [file] -Ubtdd -k -Tmfrstonebinsp+ -xpb -qmr -s0 -b2n0 -uDD -F -Sall
```

```
Static: cl+clq-wR(*,2)C :: -f [file] -Ubtdd -k -Tmfrstonemaxcsp+ -xpb -qmr -Drr -s2 -b2n0 -uDD -F -Sall
Static: cl+clq-wR(*,2)C :: -f [file] -Ubtdd -k -Tmfrstonemaxcsp+ -xpb -qmr -Drr -s3 -b2n0 -uDD -F -Sall
Static: cl+clq-wR(*,2)C :: -f [file] -Ubtdd -k -Tmfrstonemaxcsp+ -xpb -qmr -Drr -s4 -b2n0 -uDD -F -Sall
Static: cl+clq-wR(*,|C)C :: -f [file] -Ubtdd -k -Tmfrstonemaxcsp+ -xpb -qmr -s0 -b2n0 -uDD -F -Sall
```

## Finding All solutions witnessBTD:

```
GAC :: -f [file] -k -Uwbtd -uDD -DGA -F -Sall -v-1
```

```
wR(*,2)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xpb -Drr -s2 -b1 -uDD -F -Sall -v-1
wR(*,3)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xpb -Drr -s3 -b1 -uDD -F -Sall -v-1
wR(*,4)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xpb -Drr -s4 -b1 -uDD -F -Sall -v-1
```

```
Static: cl-wR(*,2)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xss -qmr -Drr -s2 -b2n0 -uDD -F -Sall
Static: cl-wR(*,3)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xss -qmr -Drr -s3 -b2n0 -uDD -F -Sall
Static: cl-wR(*,4)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xss -qmr -Drr -s4 -b2n0 -uDD -F -Sall
Static: cl-wR(*,|C|)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xss -qmr -s0 -b2n0 -uDD -F -Sall
```

```
Static: cl+proj-wR(*,2)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xpb -qmr -Drr -s2 -b2n0 -uDD -F -Sall
Static: cl+proj-wR(*,3)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xpb -qmr -Drr -s3 -b2n0 -uDD -F -Sall
Static: cl+proj-wR(*,4)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xpb -qmr -Drr -s4 -b2n0 -uDD -F -Sall
Static: cl+proj-wR(*,|C|)C :: -f [file] -Uwbtd -k -Tmfrstonenosp -xpb -qmr -s0 -b2n0 -uDD -F -Sall
```

```
Static: cl+bin-wR(*,2)C :: -f [file] -Uwbtd -k -Tmfrstonebinsp+ -xpb -qmr -Drr -s2 -b2n0 -uDD -F -Sall
Static: cl+bin-wR(*,3)C :: -f [file] -Uwbtd -k -Tmfrstonebinsp+ -xpb -qmr -Drr -s3 -b2n0 -uDD -F -Sall
Static: cl+bin-wR(*,4)C :: -f [file] -Uwbtd -k -Tmfrstonebinsp+ -xpb -qmr -Drr -s4 -b2n0 -uDD -F -Sall
Static: cl+bin-wR(*,|C|)C :: -f [file] -Uwbtd -k -Tmfrstonebinsp+ -xpb -qmr -s0 -b2n0 -uDD -F -Sall
```

```
Static: cl+clq-wR(*,2)C :: -f [file] -Uwbtd -k -Tmfrstonemaxcsp+ -xpb -qmr -Drr -s2 -b2n0 -uDD -F -Sall
Static: cl+clq-wR(*,2)C :: -f [file] -Uwbtd -k -Tmfrstonemaxcsp+ -xpb -qmr -Drr -s3 -b2n0 -uDD -F -Sall
Static: cl+clq-wR(*,2)C :: -f [file] -Uwbtd -k -Tmfrstonemaxcsp+ -xpb -qmr -Drr -s4 -b2n0 -uDD -F -Sall
Static: cl+clq-wR(*,|C|)C :: -f [file] -Uwbtd -k -Tmfrstonemaxcsp+ -xpb -qmr -s0 -b2n0 -uDD -F -Sall
```

## 4.37.2 Function Documentation

4.37.2.1 void `check_ok ( main_structure * m_s )`

4.37.2.2 void `check_ok2 ( variables * vars )`

4.37.2.3 int `main ( int argc, char ** argv )`

4.37.2.4 void `sigxcpu_handler ( int signo )`

## 4.37.3 Variable Documentation

4.37.3.1 `main_structure* m_s_g`

## 4.38 minibucket.c File Reference

```
#include "minibucket.h"
#include "rs2.h"
#include "rs.h"
#include "decomposition_functions.h"
#include "hashtable_search.h"
```

### Functions

- int `partition_bucket (decomposed_tree *tree)`
- `htable * mbe_join_tables (htable *hta, htable *htb)`

### 4.38.1 Function Documentation

4.38.1.1 `htable* mbe_join_tables ( htable * hta, htable * htb )`

4.38.1.2 `int partition_bucket ( decomposed_tree * treen )`

## 4.39 minibucket.h File Reference

```
#include "utils.h"
#include "decomposition.h"
#include "l1list.h"
```

### Functions

- `int partition_bucket (decomposed_tree *treen)`
- `int partition_bucket_old (decomposed_tree *treen)`
- `htable * mbe_join_tables (htable *hta, htable *htb)`

### 4.39.1 Detailed Description

Implements the mini-bucket algorithm.

### 4.39.2 Function Documentation

4.39.2.1 `htable* mbe_join_tables ( htable * hta, htable * htb )`

4.39.2.2 `int partition_bucket ( decomposed_tree * treen )`

4.39.2.3 `int partition_bucket_old ( decomposed_tree * treen )`

## 4.40 oneall.c File Reference

```
#include "oneall.h"
#include "utils.h"
```

### Functions

- `int evaluate_on_tree (decision_tree *dt, rs2tree_stats *rss)`
- `decision_tree * get_dt_node (char *attribute, double value, char **names)`
- `decision_tree_system * parse_tree (void)`
- `void print_decision_tree (decision_tree *dt)`

### 4.40.1 Function Documentation

4.40.1.1 `int evaluate_on_tree ( decision_tree * dt, rs2tree_stats * rss )`

4.40.1.2 `decision_tree* get_dt_node ( char * attribute, double value, char ** names )`

4.40.1.3 `decision_tree_system* parse_tree ( void )`

4.40.1.4 `void print_decision_tree ( decision_tree * dt )`

## 4.41 oneall.h File Reference

```
#include "rs2tree.h"
```

### Classes

- struct [decision\\_tree](#)
- struct [decision\\_tree\\_system](#)

### Typedefs

- typedef struct [decision\\_tree](#) [decision\\_tree](#)
- typedef struct [decision\\_tree\\_system](#) [decision\\_tree\\_system](#)

### Functions

- int [evaluate\\_on\\_tree](#) ([decision\\_tree](#) \*dt, [rs2tree\\_stats](#) \*rss)
- [decision\\_tree](#) \* [get\\_dt\\_node](#) (char \*attribute, double value, char \*\*names)
- [decision\\_tree\\_system](#) \* [parse\\_tree](#) (void)
- void [print\\_decision\\_tree](#) ([decision\\_tree](#) \*dt)

#### 4.41.1 Detailed Description

Provides the algorithm for parsing a decision tree for choosing between PerTuple and AllSol algorithms.

#### 4.41.2 Typedef Documentation

4.41.2.1 typedef struct [decision\\_tree](#) [decision\\_tree](#)

4.41.2.2 typedef struct [decision\\_tree\\_system](#) [decision\\_tree\\_system](#)

#### 4.41.3 Function Documentation

4.41.3.1 int [evaluate\\_on\\_tree](#) ( [decision\\_tree](#) \* dt, [rs2tree\\_stats](#) \* rss )

4.41.3.2 [decision\\_tree](#)\* [get\\_dt\\_node](#) ( char \* attribute, double value, char \*\* names )

4.41.3.3 [decision\\_tree\\_system](#)\* [parse\\_tree](#) ( void )

4.41.3.4 void [print\\_decision\\_tree](#) ( [decision\\_tree](#) \* dt )

## 4.42 parsers.c File Reference

```
#include "parsers.h"
#include "query_graph.h"
#include "constraints.h"
#include "decomposition.h"
#include "decomposition_functions.h"
#include <math.h>
```

## Functions

- void `read_uai` (FILE \*file, `main_structure` \*main\_struct, double weight\_th)
- void `read_ergos` (FILE \*file, `main_structure` \*main\_struct, double weight\_th)
- void `unrank` (int \*vars, int \*tuple, int rank)
- void `read_comment` (FILE \*file)
- void `read_wcsp` (FILE \*file, `main_structure` \*main\_struct)

### 4.42.1 Function Documentation

4.42.1.1 void `read_comment` ( FILE \* *file* )

4.42.1.2 void `read_ergos` ( FILE \* *file*, `main_structure` \* *main\_struct*, double *weight.th* )

4.42.1.3 void `read_uai` ( FILE \* *file*, `main_structure` \* *main\_struct*, double *weight.th* )

4.42.1.4 void `read_wcsp` ( FILE \* *file*, `main_structure` \* *main\_struct* )

4.42.1.5 void `unrank` ( int \* *vars*, int \* *tuple*, int *rank* )

## 4.43 parsers.h File Reference

```
#include "utils.h"
#include "stdio.h"
```

## Functions

- void `read_uai` (FILE \*file, `main_structure` \*main\_struct, double weight\_th)
- void `read_ergos` (FILE \*file, `main_structure` \*main\_struct, double weight\_th)
- void `read_wcsp` (FILE \*file, `main_structure` \*main\_struct)
- void `read_comment` (FILE \*file)
- void `unrank` (int \*vars, int \*tuple, int rank)

### 4.43.1 Detailed Description

Implements parsers for various CSP representations.

### 4.43.2 Function Documentation

4.43.2.1 void `read_comment` ( FILE \* *file* )

4.43.2.2 void `read_ergos` ( FILE \* *file*, `main_structure` \* *main\_struct*, double *weight.th* )

4.43.2.3 void `read_uai` ( FILE \* *file*, `main_structure` \* *main\_struct*, double *weight.th* )

4.43.2.4 void `read_wcsp` ( FILE \* *file*, `main_structure` \* *main\_struct* )

4.43.2.5 void `unrank` ( int \* *vars*, int \* *tuple*, int *rank* )



## 4.44 partition.c File Reference

```
#include "partition.h"
#include "utils.h"
#include "decomposition.h"
#include "l1ist.h"
#include "hashtable_search.h"
```

### Functions

- `l1ist * partition_separator (decomposed_tree *tree, int *varmap)`
- `l1ist * binary_rel_separator (decomposed_tree *tree, int *varmap)`
- `l1ist * all_binary_rel_separator (decomposed_tree *tree, int *varmap)`

### 4.44.1 Function Documentation

4.44.1.1 `l1ist* all_binary_rel_separator ( decomposed_tree * tree, int * varmap )`

4.44.1.2 `l1ist* binary_rel_separator ( decomposed_tree * tree, int * varmap )`

4.44.1.3 `l1ist* partition_separator ( decomposed_tree * tree, int * varmap )`

## 4.45 partition.h File Reference

```
#include "l1ist.h"
#include "decomposition.h"
```

### Functions

- `l1ist * partition_separator (decomposed_tree *tree, int *varmap)`
- `l1ist * binary_rel_separator (decomposed_tree *tree, int *varmap)`
- `l1ist * all_binary_rel_separator (decomposed_tree *tree, int *varmap)`

### 4.45.1 Detailed Description

Implements helper methods for generating redundant constraints at the separators.

### 4.45.2 Function Documentation

4.45.2.1 `l1ist* all_binary_rel_separator ( decomposed_tree * tree, int * varmap )`

4.45.2.2 `l1ist* binary_rel_separator ( decomposed_tree * tree, int * varmap )`

4.45.2.3 `l1ist* partition_separator ( decomposed_tree * tree, int * varmap )`

## 4.46 query\_graph.c File Reference

```
#include <stdio.h>
#include "query_graph.h"
#include "l1list.h"
#include "utils.h"
#include "tree_decomp.h"
#include "rs.h"
#include "rs2.h"
#include "rpwc.h"
#include "tree.h"
#include "constraints.h"
#include "relations.h"
#include "rpwcprocedures.h"
#include "rsprocedures.h"
#include "sort.h"
#include "combinations.h"
#include "db_algo1.h"
#include "rr.h"
```

### Functions

- void [delete\\_node](#) ()
- [constraint\\_graph](#) \* [new\\_constraint\\_graph](#) (int cons\_count)
- void [destroy\\_constraint\\_graph](#) ([constraint\\_graph](#) \*cg)
- [constraint\\_graph\\_node](#) \* [new\\_constraint\\_graph\\_node\\_t](#) ([htable](#) \*ht)
- [constraint\\_graph\\_node](#) \* [new\\_constraint\\_graph\\_node](#) ([constraint](#) \*cons)
- void [destroy\\_constraint\\_graph\\_node](#) ([constraint\\_graph\\_node](#) \*cgn)
- [constraint\\_graph\\_edge](#) \* [new\\_constraint\\_graph\\_edge](#) ([constraint\\_graph\\_node](#) \*node, [constraint\\_graph\\_node](#) \*neighbour, [constraint\\_graph](#) \*cg)
- void [destroy\\_constraint\\_graph\\_edge](#) ([constraint\\_graph\\_edge](#) \*cge)
- [constraint\\_graph\\_node](#) \* [get\\_other\\_end](#) ([constraint\\_graph\\_node](#) \*node, [constraint\\_graph\\_edge](#) \*edge)
- [constraint\\_graph\\_edge](#) \* [get\\_edge](#) ([constraint\\_graph\\_node](#) \*node1, [constraint\\_graph\\_node](#) \*node2, [constraint\\_graph](#) \*cg)
- [constraint\\_graph\\_node](#) \* [get\\_constraint\\_graph\\_node](#) ([constraint](#) \*cons, [constraint\\_graph](#) \*c\_graph)
- [constraint\\_graph\\_node](#) \* [get\\_node\\_with\\_constraint\\_id](#) (int id, [constraint\\_graph](#) \*cg)
- [constraint\\_graph\\_node](#) \* [add\\_constraint\\_to\\_graph](#) ([constraint](#) \*cons, [constraint\\_graph](#) \*c\_graph)
- [constraint\\_graph\\_node](#) \* [add\\_constraint\\_to\\_graph\\_force](#) ([htable](#) \*ht, [constraint\\_graph](#) \*c\_graph)
- [constraint\\_graph\\_node](#) \*\* [expand\\_cgn\\_i](#) ([constraint\\_graph](#) \*c\_graph)
- [constraint\\_graph\\_node](#) \* [add\\_or\\_get\\_constraint\\_node](#) ([constraint](#) \*cons, [constraint\\_graph](#) \*c\_graph)
- int [get\\_value\\_edge\\_sort](#) (void \*e, int row, int column)
- void \* [get\\_object\\_edge\\_sort](#) (void \*e, int row, int column)
- void [set\\_object\\_edge\\_sort](#) (void \*e, int row, int column, void \*value)
- char \*\* [build\\_adjacency\\_matrix](#) ([constraint\\_graph](#) \*cg)
- void [build\\_adjacency\\_matrix2](#) ([constraint\\_graph](#) \*cg)
- [constraint\\_graph](#) \* [init\\_graph](#) ([main\\_structure](#) \*m\_s)
- [constraint\\_graph](#) \* [init\\_graph\\_with\\_fill\\_edges](#) ([main\\_structure](#) \*m\_s, [constraints](#) \*cons, [l1list](#) \*fill\_edges, int silent)
- int [compare\\_scope\\_cge\\_global\\_list](#) (const void \*o1, const void \*o2)
- int [compare\\_scope\\_cge](#) (const void \*o1, const void \*o2)
- [constraint\\_graph](#) \* [init\\_graph\\_given\\_constraint\\_set](#) ([main\\_structure](#) \*m\_s, [set](#) \*constraints, int silent)
- void [add\\_neighbours\\_for\\_node](#) ([constraint\\_graph\\_node](#) \*c\_node, [constraint](#) \*seled\_const, [constraint\\_graph](#) \*c\_graph, [main\\_structure](#) \*m\_s)

- void `add_neighbours_for_node_given_constraint_set` (`constraint_graph_node` \*c\_node, `constraint` \*seled\_const, `constraint_graph` \*c\_graph, `set` \*constraints, `main_structure` \*m\_s)
- void `add_neighbours_for_fill_edge` (`int` \*fill\_edge, `constraint_graph` \*c\_graph, `main_structure` \*m\_s)
- void `add_neighbours_for_fill_edge_given_constraints_set` (`int` \*fill\_edge, `constraint_graph` \*c\_graph, `set` \*constraints, `main_structure` \*m\_s)
- void `check_connected` (`constraint_graph_node` \*initial\_node, `int` \*constraints, `int` \*variables)
- void `set_weights` (`constraint_graph` \*c\_g)
- void `print_number_of_deleted_tuples` (`constraint_graph` \*g)
- void `print_graph` (`constraint_graph` \*g, `int` print\_relations)
- void `print_graph_scopes` (`constraint_graph` \*g, `int` print\_relations)
- void `print_tree` (`constraint_graph_node` \*root, `int` indent)
- void `reset_colors` (`constraint_graph` \*g)
- `constraint_graph_node` \* `get_initial_node` (`constraint_graph` \*g)
- `int` `measure` (`constraint_graph_node` \*to)
- void `update_constraint_tuples` (`constraints` \*cons)
- void `evaluate_edges` (`constraint_graph` \*cg, `int` limit)
- `int` `compare` (`void` \*value, `void` \*bodyb)
- `int` `get_cnode_id` (`void` \*obj)
- `int` `merge_nodes` (`constraint_graph_node` \*nodea, `constraint_graph_node` \*nodeb, `constraint_graph` \*cg, `int` limit)
- void `remove_node_from_graph` (`constraint_graph_node` \*node, `constraint_graph` \*cg)
- void `remove_edge_from_graph` (`constraint_graph_edge` \*edge, `constraint_graph` \*cg)
- `q_node` \* `build_plan` (`constraint_graph_node` \*root)
- `int` `get_estimated_value_weight` (`void` \*body)
- void `merge_nodes_no_join` (`constraint_graph_node` \*nodea, `constraint_graph_node` \*nodeb, `constraint_graph` \*cg)
- `int` `has_common_cols` (`int` \*var1, `int` \*var2, `int` \*vmap)
- void `append_to_nodes_combination` (`constraint_graph` \*cg, `constraint_graph` \*notin\_cg, `llist` \*toadd)
- `int` `get_all_nodes_combination` (`constraint_graph` \*cg, `llist` \*combs)
- void `printGraphNeato` (`constraint_graph` \*cg, `const char` \*fileName)
- void `printGraphNeato2` (`constraint_graph` \*cg)
- `constraint_graph` \* `add_constraints_to_graph` (`llist` \*list\_of\_binary\_constraint\_vars, `constraint_graph` \*cg)
- void `remove_fill_edges_from_variables` (`variables` \*vars)
- `rbt_set` \* `new_neighbour_set` (`int` constraint\_count, `int` minimal)

## Variables

- `main_structure` \* `m_s_g`

### 4.46.1 Function Documentation

4.46.1.1 `constraint_graph_node`\* `add_constraint_to_graph` ( `constraint` \* *cons*, `constraint_graph` \* *c\_graph* )

4.46.1.2 `constraint_graph_node`\* `add_constraint_to_graph_force` ( `htable` \* *ht*, `constraint_graph` \* *c\_graph* )

4.46.1.3 `constraint_graph`\* `add_constraints_to_graph` ( `llist` \* *list\_of\_binary\_constraint\_vars*, `constraint_graph` \* *cg* )

4.46.1.4 `void` `add_neighbours_for_fill_edge` ( `int` \* *fill\_edge*, `constraint_graph` \* *c\_graph*, `main_structure` \* *m\_s* )

4.46.1.5 `void` `add_neighbours_for_fill_edge_given_constraints_set` ( `int` \* *fill\_edge*, `constraint_graph` \* *c\_graph*, `set` \* *constraints*, `main_structure` \* *m\_s* )

4.46.1.6 `void` `add_neighbours_for_node` ( `constraint_graph_node` \* *c\_node*, `constraint` \* *seled\_const*, `constraint_graph` \* *c\_graph*, `main_structure` \* *m\_s* )

- 4.46.1.7 void add\_neighbours\_for\_node\_given\_constraint\_set ( constraint\_graph\_node \* *c\_node*, constraint \* *seled\_const*, constraint\_graph \* *c\_graph*, set \* *constraints*, main\_structure \* *m\_s* )
- 4.46.1.8 constraint\_graph\_node\* add\_or\_get\_constraint\_node ( constraint \* *cons*, constraint\_graph \* *c\_graph* )
- 4.46.1.9 void append\_to\_nodes\_combination ( constraint\_graph \* *cg*, constraint\_graph \* *notin\_cg*, llist \* *toadd* )
- 4.46.1.10 char\*\* build\_adjacency\_matrix ( constraint\_graph \* *cg* )
- 4.46.1.11 void build\_adjacency\_matrix2 ( constraint\_graph \* *cg* )
- 4.46.1.12 q\_node\* build\_plan ( constraint\_graph\_node \* *root* )
- 4.46.1.13 void check\_connected ( constraint\_graph\_node \* *initial\_node*, int \* *constraints*, int \* *variables* )
- 4.46.1.14 int compare ( void \* *value*, void \* *bodyb* )
- 4.46.1.15 int compare\_scope\_cge ( const void \* *o1*, const void \* *o2* )
- 4.46.1.16 int compare\_scope\_cge\_global\_list ( const void \* *o1*, const void \* *o2* )
- 4.46.1.17 void delete\_node ( )
- 4.46.1.18 void destroy\_constraint\_graph ( constraint\_graph \* *cg* )
- 4.46.1.19 void destroy\_constraint\_graph\_edge ( constraint\_graph\_edge \* *cge* )
- 4.46.1.20 void destroy\_constraint\_graph\_node ( constraint\_graph\_node \* *cgn* )
- 4.46.1.21 void evaluate\_edges ( constraint\_graph \* *cg*, int *limit* )
- 4.46.1.22 constraint\_graph\_node\*\* expand\_cgn\_i ( constraint\_graph \* *c\_graph* )
- 4.46.1.23 int get\_all\_nodes\_combination ( constraint\_graph \* *cg*, llist \* *combs* )
- 4.46.1.24 int get\_cnode\_id ( void \* *obj* )
- 4.46.1.25 constraint\_graph\_node\* get\_constraint\_graph\_node ( constraint \* *cons*, constraint\_graph \* *c\_graph* )
- 4.46.1.26 constraint\_graph\_edge\* get\_edge ( constraint\_graph\_node \* *node1*, constraint\_graph\_node \* *node2*, constraint\_graph \* *cg* )
- 4.46.1.27 int get\_estimated\_value\_weight ( void \* *body* )
- 4.46.1.28 constraint\_graph\_node\* get\_initial\_node ( constraint\_graph \* *g* )
- 4.46.1.29 constraint\_graph\_node\* get\_node\_with\_constraint\_id ( int *id*, constraint\_graph \* *cg* )
- 4.46.1.30 void\* get\_object\_edge\_sort ( void \* *e*, int *row*, int *column* )
- 4.46.1.31 constraint\_graph\_node\* get\_other\_end ( constraint\_graph\_node \* *node*, constraint\_graph\_edge \* *edge* )
- 4.46.1.32 int get\_value\_edge\_sort ( void \* *e*, int *row*, int *column* )
- 4.46.1.33 int has\_common\_cols ( int \* *var1*, int \* *var2*, int \* *vmap* )

- 4.46.1.34 `constraint_graph*` `init_graph ( main_structure * m_s )`
- 4.46.1.35 `constraint_graph*` `init_graph_given_constraint_set ( main_structure * m_s, set * constraints, int silent )`
- 4.46.1.36 `constraint_graph*` `init_graph_with_fill_edges ( main_structure * m_s, constraints * cons, llist * fill_edges, int silent )`
- 4.46.1.37 `int` `measure ( constraint_graph_node * to )`
- 4.46.1.38 `int` `merge_nodes ( constraint_graph_node * nodea, constraint_graph_node * nodeb, constraint_graph * cg, int limit )`
- 4.46.1.39 `void` `merge_nodes_no_join ( constraint_graph_node * nodea, constraint_graph_node * nodeb, constraint_graph * cg )`
- 4.46.1.40 `constraint_graph*` `new_constraint_graph ( int cons_count )`
- 4.46.1.41 `constraint_graph_edge*` `new_constraint_graph_edge ( constraint_graph_node * node, constraint_graph_node * neighbour, constraint_graph * cg )`
- 4.46.1.42 `constraint_graph_node*` `new_constraint_graph_node ( constraint * cons )`
- 4.46.1.43 `constraint_graph_node*` `new_constraint_graph_node.t ( htable * ht )`
- 4.46.1.44 `rbt_set*` `new_neighbour_set ( int constraint_count, int minimal )`
- 4.46.1.45 `void` `print_graph ( constraint_graph * g, int print_relations )`
- 4.46.1.46 `void` `print_graph_scopes ( constraint_graph * g, int print_relations )`
- 4.46.1.47 `void` `print_number_of_deleted_tuples ( constraint_graph * g )`
- 4.46.1.48 `void` `print_tree ( constraint_graph_node * root, int indent )`
- 4.46.1.49 `void` `printGraphNeato ( constraint_graph * cg, const char * fileName )`
- 4.46.1.50 `void` `printGraphNeato2 ( constraint_graph * cg )`
- 4.46.1.51 `void` `remove_edge_from_graph ( constraint_graph_edge * edge, constraint_graph * cg )`
- 4.46.1.52 `void` `remove_fill_edges_from_variables ( variables * vars )`
- 4.46.1.53 `void` `remove_node_from_graph ( constraint_graph_node * node, constraint_graph * cg )`
- 4.46.1.54 `void` `reset_colors ( constraint_graph * g )`
- 4.46.1.55 `void` `set_object_edge_sort ( void * e, int row, int column, void * value )`
- 4.46.1.56 `void` `set_weights ( constraint_graph * c_g )`
- 4.46.1.57 `void` `update_constraint_tuples ( constraints * cons )`

## 4.46.2 Variable Documentation

- 4.46.2.1 `main_structure*` `m_s_g`

## 4.47 query\_graph.h File Reference

```
#include "globals.h"
#include "constraints.h"
#include "variables.h"
#include "hashtable.h"
#include "query_plan.h"
#include "llist.h"
#include "rbt_set.h"
```

### Classes

- struct [constraint\\_graph\\_node](#)
- struct [constraint\\_graph\\_edge](#)
- struct [constraint\\_graph](#)

### Macros

- `#define INITIAL_LIMIT 100`

### Typedefs

- typedef struct [constraint\\_graph\\_node](#) [constraint\\_graph\\_node](#)
- typedef struct [constraint\\_graph\\_edge](#) [constraint\\_graph\\_edge](#)
- typedef struct [constraint\\_graph](#) [constraint\\_graph](#)

### Functions

- char \*\* [build\\_adjacency\\_matrix](#) ([constraint\\_graph](#) \*cg)
- void [build\\_adjacency\\_matrix2](#) ([constraint\\_graph](#) \*cg)
- void [destroy\\_constraint\\_graph](#) ([constraint\\_graph](#) \*cg)
- void [destroy\\_constraint\\_graph\\_node](#) ([constraint\\_graph\\_node](#) \*cgn)
- void [destroy\\_constraint\\_graph\\_edge](#) ([constraint\\_graph\\_edge](#) \*cge)
- [constraint\\_graph\\_node](#) \*\* [expand\\_cgn\\_i](#) ([constraint\\_graph](#) \*c\_graph)
- void [check\\_connected](#) ([constraint\\_graph\\_node](#) \*initial\_node, int \*constrinats, int \*variables)
- [constraint\\_graph](#) \* [new\\_constraint\\_graph](#) (int cons\_count)
- [constraint\\_graph\\_node](#) \* [new\\_constraint\\_graph\\_node](#) ([constraint](#) \*cons)
- [constraint\\_graph\\_edge](#) \* [new\\_constraint\\_graph\\_edge](#) ([constraint\\_graph\\_node](#) \*node, [constraint\\_graph\\_node](#) \*neighbour, [constraint\\_graph](#) \*cg)
- [constraint\\_graph\\_node](#) \* [get\\_other\\_end](#) ([constraint\\_graph\\_node](#) \*node, [constraint\\_graph\\_edge](#) \*edge)
- [constraint\\_graph\\_edge](#) \* [get\\_edge](#) ([constraint\\_graph\\_node](#) \*node1, [constraint\\_graph\\_node](#) \*node2, [constraint\\_graph](#) \*cg)
- [constraint\\_graph\\_node](#) \* [get\\_constraint\\_graph\\_node](#) ([constraint](#) \*cons, [constraint\\_graph](#) \*c\_graph)
- [constraint\\_graph\\_node](#) \* [add\\_constraint\\_to\\_graph](#) ([constraint](#) \*cons, [constraint\\_graph](#) \*c\_graph)
- [constraint\\_graph\\_node](#) \* [add\\_constraint\\_to\\_graph\\_force](#) ([htable](#) \*ht, [constraint\\_graph](#) \*c\_graph)
- [constraint\\_graph\\_node](#) \* [add\\_or\\_get\\_constraint\\_node](#) ([constraint](#) \*cons, [constraint\\_graph](#) \*c\_graph)
- [constraint\\_graph](#) \* [init\\_graph](#) ([main\\_structure](#) \*m\_s)
- void [set\\_weights](#) ([constraint\\_graph](#) \*c\_g)
- void [print\\_number\\_of\\_deleted\\_tuples](#) ([constraint\\_graph](#) \*g)
- void [print\\_graph](#) ([constraint\\_graph](#) \*g, int print\_relations)

- void [print\\_graph\\_scopes](#) ([constraint\\_graph](#) \*g, int print\_relations)
- void [print\\_tree](#) ([constraint\\_graph\\_node](#) \*root, int indent)
- void [reset\\_colors](#) ([constraint\\_graph](#) \*g)
- [constraint\\_graph\\_node](#) \* [get\\_initial\\_node](#) ([constraint\\_graph](#) \*g)
- int [measure](#) ([constraint\\_graph\\_node](#) \*to)
- void [make\\_tree](#) ([constraint\\_graph\\_node](#) \*initial\_node, int [measure](#)(void))
- void [greedy\\_naive](#) ([constraint\\_graph](#) \*c\_graph)
- int [do\\_db](#) ([main\\_structure](#) \*m\_s)
- void [evaluate\\_edges](#) ([constraint\\_graph](#) \*cg, int limit)
- void [update\\_constraint\\_tuples](#) ([constraints](#) \*cons)
- void [remove\\_node\\_from\\_graph](#) ([constraint\\_graph\\_node](#) \*node, [constraint\\_graph](#) \*cg)
- void [remove\\_edge\\_from\\_graph](#) ([constraint\\_graph\\_edge](#) \*edge, [constraint\\_graph](#) \*cg)
- int [get\\_value\\_edge\\_sort](#) (void \*e, int row, int column)
- void \* [get\\_object\\_edge\\_sort](#) (void \*e, int row, int column)
- void [set\\_object\\_edge\\_sort](#) (void \*e, int row, int column, void \*value)
- [constraint\\_graph\\_node](#) \* [get\\_node\\_with\\_constraint\\_id](#) (int id, [constraint\\_graph](#) \*cg)
- int [compare](#) (void \*value, void \*bodyb)
- int [merge\\_nodes](#) ([constraint\\_graph\\_node](#) \*nodea, [constraint\\_graph\\_node](#) \*nodeb, [constraint\\_graph](#) \*cg, int limit)
- q\_node \* [build\\_plan](#) ([constraint\\_graph\\_node](#) \*root)
- int [get\\_estimated\\_value\\_weight](#) (void \*body)
- q\_node \* [buld\\_join\\_tree](#) ([constraint\\_graph](#) \*c\_graph)
- void [merge\\_nodes\\_no\\_join](#) ([constraint\\_graph\\_node](#) \*nodea, [constraint\\_graph\\_node](#) \*nodeb, [constraint\\_graph](#) \*cg)
- void [delete\\_node](#) ()
- llist \* [get\\_int\\_common\\_cols](#) (int \*var1, int \*var2, int \*vmap)
- int [has\\_common\\_cols](#) (int \*var1, int \*var2, int \*vmap)
- int [get\\_all\\_nodes\\_combination](#) ([constraint\\_graph](#) \*cg, llist \*combs)
- void [append\\_to\\_nodes\\_combination](#) ([constraint\\_graph](#) \*cg, [constraint\\_graph](#) \*notin\_cg, llist \*toadd)
- void [printGraphNeato](#) ([constraint\\_graph](#) \*cg, const char \*fileName)
- void [printGraphNeato2](#) ([constraint\\_graph](#) \*cg)
- [constraint\\_graph](#) \* [init\\_graph\\_given\\_constraint\\_set](#) ([main\\_structure](#) \*m\_s, [set](#) \*constraints, int silent)
- [constraint\\_graph](#) \* [init\\_graph\\_with\\_fill\\_edges](#) ([main\\_structure](#) \*m\_s, [constraints](#) \*cons, llist \*fill\_edges, int silent)
- void [add\\_neighbours\\_for\\_node](#) ([constraint\\_graph\\_node](#) \*c\_node, [constraint](#) \*seled\_const, [constraint\\_graph](#) \*c\_graph, [main\\_structure](#) \*m\_s)
- void [add\\_neighbours\\_for\\_node\\_given\\_constraint\\_set](#) ([constraint\\_graph\\_node](#) \*c\_node, [constraint](#) \*seled\_const, [constraint\\_graph](#) \*c\_graph, [set](#) \*constraints, [main\\_structure](#) \*m\_s)
- void [add\\_neighbours\\_for\\_fill\\_edge](#) (int \*fill\_edge, [constraint\\_graph](#) \*c\_graph, [main\\_structure](#) \*m\_s)
- void [add\\_neighbours\\_for\\_fill\\_edge\\_given\\_constraints\\_set](#) (int \*fill\_edge, [constraint\\_graph](#) \*c\_graph, [set](#) \*constraints, [main\\_structure](#) \*m\_s)
- [constraint\\_graph](#) \* [add\\_constraints\\_to\\_graph](#) (llist \*list\_of\_binary\_constraint\_vars, [constraint\\_graph](#) \*cg)
- void [remove\\_fill\\_edges\\_from\\_variables](#) ([variables](#) \*vars)
- [rbt\\_set](#) \* [new\\_neighbour\\_set](#) (int constraint\_count, int minimal)
- int [get\\_cnode\\_id](#) (void \*obj)
- [constraint\\_graph\\_node](#) \* [new\\_constraint\\_graph\\_node\\_t](#) ([htable](#) \*ht)
- int [compare\\_scope\\_cge](#) (const void \*o1, const void \*o2)
- int [compare\\_scope\\_cge\\_global\\_list](#) (const void \*o1, const void \*o2)

#### 4.47.1 Detailed Description

Implements the dual graph and methods for manipulating them.

## 4.47.2 Macro Definition Documentation

4.47.2.1 `#define INITIAL_LIMIT 100`

## 4.47.3 Typedef Documentation

4.47.3.1 `typedef struct constraint_graph constraint_graph`

4.47.3.2 `typedef struct constraint_graph_edge constraint_graph_edge`

4.47.3.3 `typedef struct constraint_graph_node constraint_graph_node`

## 4.47.4 Function Documentation

4.47.4.1 `constraint_graph_node* add_constraint_to_graph ( constraint * cons, constraint_graph * c_graph )`

4.47.4.2 `constraint_graph_node* add_constraint_to_graph_force (htable * ht, constraint_graph * c_graph )`

4.47.4.3 `constraint_graph* add_constraints_to_graph ( llist * list_of_binary_constraint_vars, constraint_graph * cg )`

4.47.4.4 `void add_neighbours_for_fill_edge ( int * fill_edge, constraint_graph * c_graph, main_structure * m_s )`

4.47.4.5 `void add_neighbours_for_fill_edge_given_constraints_set ( int * fill_edge, constraint_graph * c_graph, set * constraints, main_structure * m_s )`

4.47.4.6 `void add_neighbours_for_node ( constraint_graph_node * c_node, constraint * seled_const, constraint_graph * c_graph, main_structure * m_s )`

4.47.4.7 `void add_neighbours_for_node_given_constraint_set ( constraint_graph_node * c_node, constraint * seled_const, constraint_graph * c_graph, set * constraints, main_structure * m_s )`

4.47.4.8 `constraint_graph_node* add_or_get_constraint_node ( constraint * cons, constraint_graph * c_graph )`

4.47.4.9 `void append_to_nodes_combination ( constraint_graph * cg, constraint_graph * notin_cg, llist * toadd )`

4.47.4.10 `char** build_adjacency_matrix ( constraint_graph * cg )`

4.47.4.11 `void build_adjacency_matrix2 ( constraint_graph * cg )`

4.47.4.12 `q_node* build_plan ( constraint_graph_node * root )`

4.47.4.13 `q_node* buld_join_tree ( constraint_graph * c_graph )`

4.47.4.14 `void check_connected ( constraint_graph_node * initial_node, int * constrinats, int * variables )`

4.47.4.15 `int compare ( void * value, void * bodyb )`

4.47.4.16 `int compare_scope_cge ( const void * o1, const void * o2 )`

4.47.4.17 `int compare_scope_cge_global_list ( const void * o1, const void * o2 )`

4.47.4.18 `void delete_node ( )`

4.47.4.19 `void destroy_constraint_graph ( constraint_graph * cg )`

4.47.4.20 `void destroy_constraint_graph_edge ( constraint_graph_edge * cge )`



- 4.47.4.21 void `destroy_constraint_graph_node` ( `constraint_graph_node * cgn` )
- 4.47.4.22 int `do_db` ( `main_structure * m_s` )
- 4.47.4.23 void `evaluate_edges` ( `constraint_graph * cg`, int `limit` )
- 4.47.4.24 `constraint_graph_node**` `expand_cgn_i` ( `constraint_graph * c_graph` )
- 4.47.4.25 int `get_all_nodes_combination` ( `constraint_graph * cg`, `llist * combs` )
- 4.47.4.26 int `get_cnode_id` ( `void * obj` )
- 4.47.4.27 `constraint_graph_node*` `get_constraint_graph_node` ( `constraint * cons`, `constraint_graph * c_graph` )
- 4.47.4.28 `constraint_graph_edge*` `get_edge` ( `constraint_graph_node * node1`, `constraint_graph_node * node2`, `constraint_graph * cg` )
- 4.47.4.29 int `get_estimated_value_weight` ( `void * body` )
- 4.47.4.30 `constraint_graph_node*` `get_initial_node` ( `constraint_graph * g` )
- 4.47.4.31 `llist*` `get_int_common_cols` ( int `var1`, int `var2`, int `vmap` )
- 4.47.4.32 `constraint_graph_node*` `get_node_with_constraint_id` ( int `id`, `constraint_graph * cg` )
- 4.47.4.33 `void*` `get_object_edge_sort` ( `void * e`, int `row`, int `column` )
- 4.47.4.34 `constraint_graph_node*` `get_other_end` ( `constraint_graph_node * node`, `constraint_graph_edge * edge` )
- 4.47.4.35 int `get_value_edge_sort` ( `void * e`, int `row`, int `column` )
- 4.47.4.36 void `greedy_naive` ( `constraint_graph * c_graph` )
- 4.47.4.37 int `has_common_cols` ( int `var1`, int `var2`, int `vmap` )
- 4.47.4.38 `constraint_graph*` `init_graph` ( `main_structure * m_s` )
- 4.47.4.39 `constraint_graph*` `init_graph_given_constraint_set` ( `main_structure * m_s`, `set * constraints`, int `silent` )
- 4.47.4.40 `constraint_graph*` `init_graph_with_fill_edges` ( `main_structure * m_s`, `constraints * cons`, `llist * fill_edges`, int `silent` )
- 4.47.4.41 void `make_tree` ( `constraint_graph_node * initial_node`, int `measurevoid` )
- 4.47.4.42 int `measure` ( `constraint_graph_node * to` )
- 4.47.4.43 int `merge_nodes` ( `constraint_graph_node * nodea`, `constraint_graph_node * nodeb`, `constraint_graph * cg`, int `limit` )
- 4.47.4.44 void `merge_nodes_no_join` ( `constraint_graph_node * nodea`, `constraint_graph_node * nodeb`, `constraint_graph * cg` )
- 4.47.4.45 `constraint_graph*` `new_constraint_graph` ( int `cons_count` )

- 4.47.4.46 `constraint_graph_edge*` `new_constraint_graph_edge ( constraint_graph_node * node, constraint_graph_node * neighbour, constraint_graph * cg )`
- 4.47.4.47 `constraint_graph_node*` `new_constraint_graph_node ( constraint * cons )`
- 4.47.4.48 `constraint_graph_node*` `new_constraint_graph_node_t ( htable * ht )`
- 4.47.4.49 `rbt_set*` `new_neighbour_set ( int constraint_count, int minimal )`
- 4.47.4.50 `void` `print_graph ( constraint_graph * g, int print_relations )`
- 4.47.4.51 `void` `print_graph_scopes ( constraint_graph * g, int print_relations )`
- 4.47.4.52 `void` `print_number_of_deleted_tuples ( constraint_graph * g )`
- 4.47.4.53 `void` `print_tree ( constraint_graph_node * root, int indent )`
- 4.47.4.54 `void` `printGraphNeato ( constraint_graph * cg, const char * fileName )`
- 4.47.4.55 `void` `printGraphNeato2 ( constraint_graph * cg )`
- 4.47.4.56 `void` `remove_edge_from_graph ( constraint_graph_edge * edge, constraint_graph * cg )`
- 4.47.4.57 `void` `remove_fill_edges_from_variables ( variables * vars )`
- 4.47.4.58 `void` `remove_node_from_graph ( constraint_graph_node * node, constraint_graph * cg )`
- 4.47.4.59 `void` `reset_colors ( constraint_graph * g )`
- 4.47.4.60 `void` `set_object_edge_sort ( void * e, int row, int column, void * value )`
- 4.47.4.61 `void` `set_weights ( constraint_graph * c.g )`
- 4.47.4.62 `void` `update_constraint_tuples ( constraints * cons )`

## 4.48 rb\_tree.c File Reference

```
#include "rb_tree.h"
```

### Functions

- `rb_tree *` `new_rb_tree (void *key, int ikey, void *payload)`
- `void` `destroy_rb_tree (rb_tree **root)`
- `void` `destroy_rb_tree_key (rb_tree **root, void destroyer(llist *))`
- `void` `destroy_rb_tree_paload (rb_tree **root, void destroyer(llist *))`
- `void` `rb_left_rotate (rb_tree **root, rb_tree *x)`
- `void` `rb_right_rotate (rb_tree **root, rb_tree *x)`
- `void *` `rb_insert (rb_tree **root, void *key, int ikey, void *payload, int compare(int *, int *))`
- `void` `rb_insert_fixup (rb_tree **root, rb_tree *z)`
- `void *` `rb_find (rb_tree *root, void *key, int ikey, int compare(int *, int *))`
- `void` `rb_get_sorted_order (rb_tree *root, int compare(int *, int *), llist *sorted)`
- `void` `rb_get_sorted_order_list (rb_tree *root, int compare(int *, int *), llist *sorted, llist *keys)`
- `rb_tree *` `rb_tree_union (rb_tree *a, rb_tree *b, int compare(int *, int *))`
- `rb_tree *` `rb_copy_tree (rb_tree *tree, int compare(int *, int *))`

- void \* `rb_delete_key` (`rb_tree **root`, void \*`key`, int `ikey`, int `compare`(int \*, int \*))
- void \* `rb_delete` (`rb_tree **root`, `rb_tree *z`)
- void `rb_delete_fixup` (`rb_tree **root`, `rb_tree *x`)
- `rb_tree * rb_successor` (`rb_tree *x`)
- `rb_tree * rb_minimum` (`rb_tree *x`)
- `rb_tree * rb_maximum` (`rb_tree *x`)
- void `check_tree` (`rb_tree *root`)
- void `rb_print_tree` (`rb_tree *root`)
- void `rb_print_tree_l` (`rb_tree *root`, int `l`)

#### 4.48.1 Function Documentation

4.48.1.1 void `check_tree` ( `rb_tree * root` )

4.48.1.2 void `destroy_rb_tree` ( `rb_tree ** root` )

4.48.1.3 void `destroy_rb_tree_key` ( `rb_tree ** root`, void `destroyerlist` \* )

4.48.1.4 void `destroy_rb_tree_payload` ( `rb_tree ** root`, void `destroyerlist` \* )

4.48.1.5 `rb_tree*` `new_rb_tree` ( void \* `key`, int `ikey`, void \* `payload` )

4.48.1.6 `rb_tree*` `rb_copy_tree` ( `rb_tree * tree`, int `compareint` \*, int \* )

4.48.1.7 void\* `rb_delete` ( `rb_tree ** root`, `rb_tree * z` )

4.48.1.8 void `rb_delete_fixup` ( `rb_tree ** root`, `rb_tree * x` )

4.48.1.9 void\* `rb_delete_key` ( `rb_tree ** root`, void \* `key`, int `ikey`, int `compareint` \*, int \* )

4.48.1.10 void\* `rb_find` ( `rb_tree * root`, void \* `key`, int `ikey`, int `compareint` \*, int \* )

4.48.1.11 void `rb_get_sorted_order` ( `rb_tree * root`, int `compareint` \*, int \*, `l`list \* `sorted` )

4.48.1.12 void `rb_get_sorted_order_list` ( `rb_tree * root`, int `compareint` \*, int \*, `l`list \* `sorted`, `l`list \* `keys` )

4.48.1.13 void\* `rb_insert` ( `rb_tree ** root`, void \* `key`, int `ikey`, void \* `payload`, int `compareint` \*, int \* )

4.48.1.14 void `rb_insert_fixup` ( `rb_tree ** root`, `rb_tree * z` )

4.48.1.15 void `rb_left_rotate` ( `rb_tree ** root`, `rb_tree * x` )

4.48.1.16 `rb_tree*` `rb_maximum` ( `rb_tree * x` )

4.48.1.17 `rb_tree*` `rb_minimum` ( `rb_tree * x` )

4.48.1.18 void `rb_print_tree` ( `rb_tree * root` )

4.48.1.19 void `rb_print_tree_l` ( `rb_tree * root`, int `l` )

4.48.1.20 void `rb_right_rotate` ( `rb_tree ** root`, `rb_tree * x` )

4.48.1.21 `rb_tree*` `rb_successor` ( `rb_tree * x` )

4.48.1.22 `rb_tree*` `rb_tree_union` ( `rb_tree * a`, `rb_tree * b`, int `compareint` \*, int \* )

## 4.49 rb\_tree.h File Reference

```
#include "l1ist.h"
#include "utils.h"
```

### Classes

- struct [rb\\_tree](#)

### Typedefs

- typedef struct [rb\\_tree](#) [rb\\_tree](#)

### Functions

- void [destroy\\_rb\\_tree](#) ([rb\\_tree](#) \*\*root)
- void [destroy\\_rb\\_tree\\_key](#) ([rb\\_tree](#) \*\*root, void destroyer([l1ist](#) \*))
- void [destroy\\_rb\\_tree\\_paload](#) ([rb\\_tree](#) \*\*root, void destroyer([l1ist](#) \*))
- [rb\\_tree](#) \* [new\\_rb\\_tree](#) (void \*key, int ikey, void \*payload)
- void [rb\\_left\\_rotate](#) ([rb\\_tree](#) \*\*root, [rb\\_tree](#) \*x)
- void [rb\\_right\\_rotate](#) ([rb\\_tree](#) \*\*root, [rb\\_tree](#) \*x)
- void \* [rb\\_insert](#) ([rb\\_tree](#) \*\*root, void \*key, int ikey, void \*payload, int [compare](#)(int \*, int \*))
- void [rb\\_insert\\_fixup](#) ([rb\\_tree](#) \*\*root, [rb\\_tree](#) \*z)
- void \* [rb\\_find](#) ([rb\\_tree](#) \*root, void \*key, int ikey, int [compare](#)(int \*, int \*))
- void [rb\\_get\\_sorted\\_order](#) ([rb\\_tree](#) \*root, int [compare](#)(int \*, int \*), [l1ist](#) \*sorted)
- void [rb\\_get\\_sorted\\_order\\_list](#) ([rb\\_tree](#) \*root, int [compare](#)(int \*, int \*), [l1ist](#) \*sorted, [l1ist](#) \*keys)
- [rb\\_tree](#) \* [rb\\_copy\\_tree](#) ([rb\\_tree](#) \*tree, int [compare](#)(int \*, int \*))
- [rb\\_tree](#) \* [rb\\_successor](#) ([rb\\_tree](#) \*x)
- [rb\\_tree](#) \* [rb\\_minimum](#) ([rb\\_tree](#) \*x)
- [rb\\_tree](#) \* [rb\\_maximum](#) ([rb\\_tree](#) \*x)
- void [rb\\_delete\\_fixup](#) ([rb\\_tree](#) \*\*root, [rb\\_tree](#) \*x)
- void \* [rb\\_delete](#) ([rb\\_tree](#) \*\*root, [rb\\_tree](#) \*z)
- void \* [rb\\_delete\\_key](#) ([rb\\_tree](#) \*\*root, void \*key, int ikey, int [compare](#)(int \*, int \*))
- void [rb\\_print\\_tree\\_l](#) ([rb\\_tree](#) \*root, int l)
- void [rb\\_print\\_tree](#) ([rb\\_tree](#) \*root)
- void [check\\_tree](#) ([rb\\_tree](#) \*root)
- [rb\\_tree](#) \* [rb\\_tree\\_union](#) ([rb\\_tree](#) \*a, [rb\\_tree](#) \*b, int [compare](#)(int \*, int \*))

#### 4.49.1 Detailed Description

Provides a red-black tree implementation.

#### 4.49.2 Typedef Documentation

##### 4.49.2.1 typedef struct [rb\\_tree](#) [rb\\_tree](#)

#### 4.49.3 Function Documentation

##### 4.49.3.1 void [check\\_tree](#) ( [rb\\_tree](#) \* root )

- 4.49.3.2 void `destroy_rb_tree` ( `rb_tree ** root` )
- 4.49.3.3 void `destroy_rb_tree_key` ( `rb_tree ** root`, void `destroyerl` )
- 4.49.3.4 void `destroy_rb_tree_payload` ( `rb_tree ** root`, void `destroyerl` )
- 4.49.3.5 `rb_tree*` `new_rb_tree` ( void `* key`, int `ikey`, void `* payload` )
- 4.49.3.6 `rb_tree*` `rb_copy_tree` ( `rb_tree * tree`, int `compareint`, int )
- 4.49.3.7 void\* `rb_delete` ( `rb_tree ** root`, `rb_tree * z` )
- 4.49.3.8 void `rb_delete_fixup` ( `rb_tree ** root`, `rb_tree * x` )
- 4.49.3.9 void\* `rb_delete_key` ( `rb_tree ** root`, void `* key`, int `ikey`, int `compareint`, int )
- 4.49.3.10 void\* `rb_find` ( `rb_tree * root`, void `* key`, int `ikey`, int `compareint`, int )
- 4.49.3.11 void `rb_get_sorted_order` ( `rb_tree * root`, int `compareint`, int, `l` )
- 4.49.3.12 void `rb_get_sorted_order_list` ( `rb_tree * root`, int `compareint`, int, `l`, `l` )
- 4.49.3.13 void\* `rb_insert` ( `rb_tree ** root`, void `* key`, int `ikey`, void `* payload`, int `compareint`, int )
- 4.49.3.14 void `rb_insert_fixup` ( `rb_tree ** root`, `rb_tree * z` )
- 4.49.3.15 void `rb_left_rotate` ( `rb_tree ** root`, `rb_tree * x` )
- 4.49.3.16 `rb_tree*` `rb_maximum` ( `rb_tree * x` )
- 4.49.3.17 `rb_tree*` `rb_minimum` ( `rb_tree * x` )
- 4.49.3.18 void `rb_print_tree` ( `rb_tree * root` )
- 4.49.3.19 void `rb_print_tree_l` ( `rb_tree * root`, int `l` )
- 4.49.3.20 void `rb_right_rotate` ( `rb_tree ** root`, `rb_tree * x` )
- 4.49.3.21 `rb_tree*` `rb_successor` ( `rb_tree * x` )
- 4.49.3.22 `rb_tree*` `rb_tree_union` ( `rb_tree * a`, `rb_tree * b`, int `compareint`, int )

## 4.50 rbt\_set.c File Reference

```
#include "rbt_set.h"
#include "utils.h"
```

### Functions

- `rbt_set *` `new_rbt_set` (int size)
- int `add_element_rbt` (`rbt_set *s`, void `*e`, int key)
- int `set_empty_rbt` (`rbt_set *s`)
- void `destroy_set_rbt` (`rbt_set *s`)
- void `* remove_element_with_key_rbt` (`rbt_set *s`, int key)

- void \* [get\\_element\\_with\\_key\\_rbt](#) (rbt\_set \*s, int key)
- int [remove\\_int\\_element\\_rbt](#) (rbt\_set \*s, int key)
- int [get\\_int\\_element\\_rbt](#) (rbt\_set \*s, int key)
- int [add\\_int\\_element\\_rbt](#) (rbt\_set \*s, int key)
- rbt\_set \* [copy\\_set\\_rbt](#) (rbt\_set \*s)
- void [reset\\_set\\_rbt](#) (rbt\_set \*s)
- llist \* [set\\_to\\_list\\_rbt](#) (rbt\_set \*s)

#### 4.50.1 Function Documentation

4.50.1.1 int [add\\_element\\_rbt](#) ( rbt\_set \* s, void \* e, int key )

4.50.1.2 int [add\\_int\\_element\\_rbt](#) ( rbt\_set \* s, int key )

4.50.1.3 rbt\_set\* [copy\\_set\\_rbt](#) ( rbt\_set \* s )

4.50.1.4 void [destroy\\_set\\_rbt](#) ( rbt\_set \* s )

4.50.1.5 void\* [get\\_element\\_with\\_key\\_rbt](#) ( rbt\_set \* s, int key )

4.50.1.6 int [get\\_int\\_element\\_rbt](#) ( rbt\_set \* s, int key )

4.50.1.7 rbt\_set\* [new\\_rbt\\_set](#) ( int size )

4.50.1.8 void\* [remove\\_element\\_with\\_key\\_rbt](#) ( rbt\_set \* s, int key )

4.50.1.9 int [remove\\_int\\_element\\_rbt](#) ( rbt\_set \* s, int key )

4.50.1.10 void [reset\\_set\\_rbt](#) ( rbt\_set \* s )

4.50.1.11 int [set\\_empty\\_rbt](#) ( rbt\_set \* s )

4.50.1.12 llist\* [set\\_to\\_list\\_rbt](#) ( rbt\_set \* s )

### 4.51 rbt\_set.h File Reference

```
#include "set.h"
#include "rb_tree.h"
```

#### Classes

- struct [rbt\\_set](#)

#### Typedefs

- typedef struct [rbt\\_set](#) rbt\_set

#### Functions

- rbt\_set \* [new\\_rbt\\_set](#) (int size)
- int [add\\_element\\_rbt](#) (rbt\_set \*s, void \*e, int key)
- int [set\\_empty\\_rbt](#) (rbt\_set \*s)

- void `destroy_set_rbt` (`rbt_set *s`)
- void \* `peek_element_rbt` (`rbt_set *s`)
- void \* `remove_element_with_key_rbt` (`rbt_set *s`, `int key`)
- void \* `get_element_with_key_rbt` (`rbt_set *s`, `int key`)
- int `remove_int_element_rbt` (`rbt_set *s`, `int key`)
- int `get_int_element_rbt` (`rbt_set *s`, `int key`)
- int `add_int_element_rbt` (`rbt_set *s`, `int key`)
- `rbt_set *` `copy_set_rbt` (`rbt_set *s`)
- void `reset_set_rbt` (`rbt_set *s`)
- `llist *` `set_to_list_rbt` (`rbt_set *s`)

### 4.51.1 Detailed Description

Provides a set implementation using red-black trees.

### 4.51.2 Typedef Documentation

4.51.2.1 typedef struct `rbt_set` `rbt_set`

### 4.51.3 Function Documentation

4.51.3.1 int `add_element_rbt` ( `rbt_set * s`, `void * e`, `int key` )

4.51.3.2 int `add_int_element_rbt` ( `rbt_set * s`, `int key` )

4.51.3.3 `rbt_set*` `copy_set_rbt` ( `rbt_set * s` )

4.51.3.4 void `destroy_set_rbt` ( `rbt_set * s` )

4.51.3.5 void\* `get_element_with_key_rbt` ( `rbt_set * s`, `int key` )

4.51.3.6 int `get_int_element_rbt` ( `rbt_set * s`, `int key` )

4.51.3.7 `rbt_set*` `new_rbt_set` ( `int size` )

4.51.3.8 void\* `peek_element_rbt` ( `rbt_set * s` )

4.51.3.9 void\* `remove_element_with_key_rbt` ( `rbt_set * s`, `int key` )

4.51.3.10 int `remove_int_element_rbt` ( `rbt_set * s`, `int key` )

4.51.3.11 void `reset_set_rbt` ( `rbt_set * s` )

4.51.3.12 int `set_empty_rbt` ( `rbt_set * s` )

4.51.3.13 `llist*` `set_to_list_rbt` ( `rbt_set * s` )

## 4.52 rpwc.c File Reference

```
#include "rpwc.h"
#include "utils.h"
```

## Functions

- `lastGAC * new_lastGAC (htable *ht, int varid)`
- `void destroy_lastGAC (lastGAC *lg)`
- `undo_lastGAC * new_undo_lastGAC (void)`
- `undo_info * new_undo_info (void)`
- `void destroy_undo_info (undo_info *ui)`
- `void init_rpwc (variables *vars)`
- `tripplet * new_tripplet (int a, int b, int c)`
- `int reviseMaxRPWC (variable *var, constraint *cons, llist **reduct, undo_lastGAC **undo_LG)`
- `int reviseGAC (variable *var, constraint *cons, llist **reduct, undo_lastGAC **undo_LG)`
- `int reviseGACMaxRPWC (variable *var, constraint *cons, llist **reduct, undo_lastGAC **undo_LG, int algo)`
- `int domain_filter (variable *c_var)`
- `void enqueue (variable *var, constraint *ci, set *queue)`

### 4.52.1 Function Documentation

4.52.1.1 `void destroy_lastGAC ( lastGAC * lg )`

4.52.1.2 `void destroy_undo_info ( undo_info * ui )`

4.52.1.3 `int domain_filter ( variable * c_var )`

4.52.1.4 `void enqueue ( variable * var, constraint * ci, set * queue )`

4.52.1.5 `void init_rpwc ( variables * vars )`

4.52.1.6 `lastGAC* new_lastGAC ( htable * ht, int varid )`

4.52.1.7 `tripplet* new_tripplet ( int a, int b, int c )`

4.52.1.8 `undo_info* new_undo_info ( void )`

4.52.1.9 `undo_lastGAC* new_undo_lastGAC ( void )`

4.52.1.10 `int reviseGAC ( variable * var, constraint * cons, llist ** reduct, undo_lastGAC ** undo_LG )`

4.52.1.11 `int reviseGACMaxRPWC ( variable * var, constraint * cons, llist ** reduct, undo_lastGAC ** undo_LG, int algo )`

4.52.1.12 `int reviseMaxRPWC ( variable * var, constraint * cons, llist ** reduct, undo_lastGAC ** undo_LG )`

### 4.53 rpwc.h File Reference

```
#include "variables.h"
#include "hashtable.h"
#include "llist.h"
#include "domains.h"
#include "globals.h"
#include "query_graph.h"
#include "set.h"
#include "utils.h"
```



## Classes

- struct [lastGAC](#)
- struct [undo\\_lastGAC](#)
- struct [tripplet](#)
- struct [undo\\_info](#)

## Typedefs

- typedef struct [lastGAC](#) [lastGAC](#)
- typedef struct [undo\\_lastGAC](#) [undo\\_lastGAC](#)
- typedef struct [tripplet](#) [tripplet](#)
- typedef struct [undo\\_info](#) [undo\\_info](#)

## Enumerations

- enum [reviseAlgo](#) { [NONE](#), [GAC](#), [maxRPWC](#) }

## Functions

- [undo\\_lastGAC](#) \* [new\\_undo\\_lastGAC](#) (void)
- [tripplet](#) \* [new\\_tripplet](#) (int a, int b, int c)
- [undo\\_info](#) \* [new\\_undo\\_info](#) (void)
- void [destroy\\_undo\\_info](#) ([undo\\_info](#) \*ui)
- [lastGAC](#) \* [new\\_lastGAC](#) ([htable](#) \*ht, int varid)
- void [destroy\\_lastGAC](#) ([lastGAC](#) \*lg)
- void [init\\_rpwc](#) ([variables](#) \*vars)
- int [reviseGACMaxRPWC](#) ([variable](#) \*var, [constraint](#) \*cons, [llist](#) \*\*reduct, [undo\\_lastGAC](#) \*\*undo\_LG, int algo)
- int [reviseMaxRPWC](#) ([variable](#) \*var, [constraint](#) \*cons, [llist](#) \*\*reduct, [undo\\_lastGAC](#) \*\*undo\_LG)
- int [reviseGAC](#) ([variable](#) \*var, [constraint](#) \*cons, [llist](#) \*\*reduct, [undo\\_lastGAC](#) \*\*undo\_LG)
- int [domain\\_filter](#) ([variable](#) \*c\_var)
- void [enqueue](#) ([variable](#) \*var, [constraint](#) \*ci, [set](#) \*queue)

### 4.53.1 Detailed Description

Provides the implementations for the GAC and maxRPWC algorithms.

### 4.53.2 Typedef Documentation

#### 4.53.2.1 typedef struct lastGAC lastGAC

#### 4.53.2.2 typedef struct triplet triplet

#### 4.53.2.3 typedef struct undo\_info undo\_info

#### 4.53.2.4 typedef struct undo\_lastGAC undo\_lastGAC

### 4.53.3 Enumeration Type Documentation

#### 4.53.3.1 enum reviseAlgo

Enumerator

***NONE***

**GAC****maxRPWC****4.53.4 Function Documentation**

4.53.4.1 void `destroy_lastGAC` ( `lastGAC * lg` )

4.53.4.2 void `destroy_undo_info` ( `undo_info * ui` )

4.53.4.3 int `domain_filter` ( `variable * c_var` )

4.53.4.4 void `enqueue` ( `variable * var`, `constraint * ci`, `set * queue` )

4.53.4.5 void `init_rpwC` ( `variables * vars` )

4.53.4.6 `lastGAC*` `new_lastGAC` ( `htable * ht`, `int varid` )

4.53.4.7 `tripplet*` `new_tripplet` ( `int a`, `int b`, `int c` )

4.53.4.8 `undo_info*` `new_undo_info` ( `void` )

4.53.4.9 `undo_lastGAC*` `new_undo_lastGAC` ( `void` )

4.53.4.10 int `reviseGAC` ( `variable * var`, `constraint * cons`, `lList ** reduct`, `undo_lastGAC ** undo_LG` )

4.53.4.11 int `reviseGACMaxRPWC` ( `variable * var`, `constraint * cons`, `lList ** reduct`, `undo_lastGAC ** undo_LG`, `int algo` )

4.53.4.12 int `reviseMaxRPWC` ( `variable * var`, `constraint * cons`, `lList ** reduct`, `undo_lastGAC ** undo_LG` )

**4.54 rr.c File Reference**

```
#include <stdlib.h>
#include <stdio.h>
#include "utils.h"
#include "lList.h"
#include "set.h"
#include "rr.h"
#include "rb_tree.h"
#include "tables.h"
```

**Functions**

- `constraint_graph * rr` (`constraint_graph *cg`, `int cons_last_id`)
- `rb_tree * get_overlaps` (`constraint_graph *cg`)
- `lList * get_relation_groups` (`lList *sorted_overlaps`, `lList *sorted_overlaps_keys`)
- void `add_rr_edges` (`lList *connected_components`, `constraint_graph *g_sub_k`)

**4.54.1 Function Documentation**

4.54.1.1 void `add_rr_edges` ( `lList * connected_components`, `constraint_graph * g_sub_k` )

4.54.1.2 `rb_tree* get_overlaps ( constraint_graph * cg )`

4.54.1.3 `llist* get_relation_groups ( llist * sorted_overlaps, llist * sorted_overlaps_keys )`

4.54.1.4 `constraint_graph* rr ( constraint_graph * cg, int cons_last_id )`

## 4.55 rr.h File Reference

```
#include "query_graph.h"
#include "constraints.h"
#include "rb_tree.h"
```

### Functions

- `constraint_graph * rr (constraint_graph *cg, int cons_last_id)`
- `rb_tree * get_overlaps (constraint_graph *cg)`
- `llist * get_relation_groups (llist *overlaps, llist *sorted_overlaps_keys)`
- void `add_rr_edges (llist *connected_components, constraint_graph *g_sub_k)`

### 4.55.1 Detailed Description

Implements the redundancy-removal algorithm.

### 4.55.2 Function Documentation

4.55.2.1 void `add_rr_edges ( llist * connected_components, constraint_graph * g_sub_k )`

4.55.2.2 `rb_tree* get_overlaps ( constraint_graph * cg )`

4.55.2.3 `llist* get_relation_groups ( llist * overlaps, llist * sorted_overlaps_keys )`

4.55.2.4 `constraint_graph* rr ( constraint_graph * cg, int cons_last_id )`

## 4.56 rs2.c File Reference

```
#include "rs2.h"
#include "tree.h"
#include "utils.h"
#include "llist.h"
#include "tree_map.h"
#include "query_graph.h"
#include "rs2tree.h"
#include "oneall.h"
#include "gbtd.h"
```

### Functions

- int `stage4_filter_with_no_check_insol_tuples (set *queue, int time, set *undo_set, set *cluster_vars, light_stack *undo_stack)`

- int `stage4_filter_with_all_solutions` (set \*queue, int time, set \*undo\_set, set \*cluster\_vars, light\_stack \*undo\_stack, int \*queues\_total\_count, int ppr)
- int `stage4_filter_auto` (set \*queue, set \*\*queues, int time, set \*undo\_set, set \*cluster\_vars, light\_stack \*undo\_stack, int \*queues\_total\_count, int ppr)
- void `get_rstat` (comb\_data \*cdata)
- int `stage4_filter_auto_one_sol` (set \*queue, set \*\*queues, int time, set \*undo\_set, light\_stack \*undo\_stack, int \*queues\_total\_count)
- int `stage4_filter_auto_all_sol` (set \*queue, set \*\*queues, int time, set \*undo\_set, light\_stack \*undo\_stack, int \*queues\_total\_count)
- int `stage4_filter` (set \*queue, int time, set \*undo\_set, set \*cluster\_vars, light\_stack \*undo\_stack, int ppr)
- int `stage4_filter_multiqueue` (set \*queue, int time, set \*undo\_set, set \*cluster\_vars, light\_stack \*undo\_stack, set \*\*queues, int \*queues\_total\_count, int ppr)
- set \* `stage4_init_queue` (llist \*constraint\_graph\_node\_list)
- void `queue_relation_related` (htable \*ht, set \*queue)
- void `queue_relation_related_multiqueue` (htable \*ht, set \*queue, set \*\*queues, int \*queues\_total\_count)
- int `setup_relation_groups` (llist \*combs, int context, options category)
- int `setup_relation_groups_` (llist \*combs, int context, options category, comb\_data \*\*cdatas, llist \*context\_list)
- void `enum_table_pairs` (htable \*hta0, llist\_node \*combnp, int \*varmap, comb\_data \*cd, comb\_data \*base\_cd)
- comb\_data \* `make_comb_data2` (llist \*comb, llist\_node \*combnp, int context, int no\_tuple\_positions, options category, comb\_data \*base\_cd)
- htable \* `get_hts_paired` (comb\_data \*cd, int hi, int hj, int p)
- tree\_map \* `get_tms_paired` (comb\_data \*cd, int hi, int hj)
- llist \* `get_pair_common_cols` (comb\_data \*cd, int hi, int hj)
- void `destroy_comb_data2` (comb\_data \*cd)
- void `build_ind_trees` (comb\_data \*cd, llist\_node \*combnp, int \*varmap, comb\_data \*base\_cd)
- int `find_support_for_single_solution` (int t, comb\_data \*cd, set \*cluster\_vars, double \*sol\_count)
- int `get_revision_time` (htable \*ht, int t)
- void `set_revision_time` (htable \*ht, int t, int revision\_time, comb\_data \*cd)
- void `set_revision_time_parents` (htable \*ht, int t, int revision\_time, comb\_data \*cd, llist \*stack, int \*\*tuples)
- int `find_support_for_single_solution_bc` (int t, comb\_data \*cd)
- int `find_support_for_single_solution_bc_rec` (int \*\*tuples, comb\_data \*cd, int level, int \*tuple\_positions, int \*tuple\_position\_htid)
- int `find_support_for_single_solution_with_fc` (int t, comb\_data \*cd, int first\_table, int all\_solutions, llist \*remaining\_vars, set \*cluster\_vars, active\_tuple \*ac, double \*sol\_count)
- int `find_support_for_single_solution_with_fc_priority` (int t, comb\_data \*cd, int first\_table, int all\_solutions, llist \*remaining\_vars, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, double \*sol\_count)
- htable \* `get_ht_from_cd` (comb\_data \*cd, int ht\_pos)
- int `mark_all_in_future` (int \*old\_marks, int mark\_with, int var, llist \*remaining\_vars, int \*current\_tuple, comb\_data \*cd, llist \*undo\_dom, int \*marks, llist \*\*\*doms, int rec\_depth, int drop\_for\_all\_revised)
- void `set_deg_count` (comb\_data \*cd, llist \*remaining\_vars, llist \*undo\_deg, int removed)
- int `choose_var` (comb\_data \*cd, llist \*remaining\_vars, llist \*undo\_deg)
- int `choose_var_priority` (comb\_data \*cd, llist \*remaining\_vars, llist \*undo\_deg, int priority\_last\_n)
- int `choose_var_priority_decided` (comb\_data \*cd, llist \*remaining\_vars, llist \*undo\_deg, int priority\_last\_n, int decided, active\_tuple \*ac)
- void `set_dom` (comb\_data \*cd)
- void `undo_dom_deg_var` (llist \*undo\_deg, llist \*undo\_dom, comb\_data \*cd, int var, llist \*remaining\_vars, llist \*\*\*doms, llist \*\*doms\_xlevel)
- int `visit_rest_of_tuples` (llist \*remaining\_vars, int all\_solutions, int next\_var, comb\_data \*cd, int var, int \*old\_marks, int \*\*tuples, llist \*\*\*doms, int rec\_depth, int \*tuple\_positions, llist \*undo\_deg, llist \*undo\_dom)
- int `find_support_for_single_solution_with_fc_rec_single_out` (int rec\_depth, int \*\*allocated\_marks, int \*old\_marks, int \*\*tuples, comb\_data \*cd, int var, int \*tuple\_positions, int \*tuple\_position\_htid, llist \*remaining\_vars, int all\_solutions, llist \*\*\*doms, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, double \*sol\_count, llist \*stack)

- int [find\\_support\\_for\\_single\\_solution\\_with\\_fc\\_rec](#) (int rec\_depth, int \*\*allocated\_marks, int \*old\_marks, int \*\*tuples, comb\_data \*cd, int var, int \*tuple\_positions, int \*tuple\_position\_htid, llist \*remaining\_vars, int all\_solutions, llist \*\*doms, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, double \*sol\_count, llist \*stack)
- int [filter\\_all\\_relations\\_with\\_all\\_solutions](#) (comb\_data \*cd, set \*cluster\_vars, active\_tuple \*ac, double \*sol\_count)
- int [filter\\_all\\_relations\\_with\\_all\\_solutions\\_priority](#) (comb\_data \*cd, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, double \*sol\_count)
- void [find\\_all\\_supports\\_with\\_all\\_solutions\\_with\\_fc](#) (comb\_data \*cd, set \*cluster\_vars, active\_tuple \*ac, double \*sol\_count)
- void [find\\_all\\_supports\\_with\\_all\\_solutions\\_with\\_fc\\_priority](#) (comb\_data \*cd, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, double \*sol\_count)
- void [find\\_all\\_supports\\_with\\_all\\_solutions\\_with\\_fc\\_priority\\_next\\_var](#) (comb\_data \*cd, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, int next\_v, double \*sol\_count)
- int [filter\\_single\\_relation\\_in\\_combination](#) (comb\_data \*cd, set \*undo\_set, light\_stack \*undo\_stack, double \*sol\_count)
- comb\_data \* [setup\\_relation\\_group\\_for\\_a\\_comb](#) (llist \*comb, int context, options category, comb\_data \*base\_cd)
- comb\_data \* [setup\\_relation\\_group\\_for\\_a\\_comb\\_first](#) (llist \*comb, int context, llist\_node \*firsttable, options category, comb\_data \*base\_cd)
- comb\_data \* [setup\\_relation\\_group\\_for\\_a\\_comb\\_long\\_indexes](#) (llist \*comb, options category)
- llist \* [get\\_comb\\_from](#) (htable \*filter, htable \*given)
- void [destroy\\_comb](#) (llist \*comb)
- comb\_cdata \* [new\\_comb\\_cdata](#) (int context)
- void [destroy\\_comb\\_cdata](#) (comb\_cdata \*ccd)
- void [clear\\_comb\\_datas\\_from\\_hts](#) (llist \*htlist)

## Variables

- void \* [watch](#) = NULL

### 4.56.1 Function Documentation

4.56.1.1 void [build\\_ind\\_trees](#) ( comb\_data \* cd, llist\_node \* combnp, int \* varmap, comb\_data \* base\_cd )

4.56.1.2 int [choose\\_var](#) ( comb\_data \* cd, llist \* remaining\_vars, llist \* undo\_deg )

4.56.1.3 int [choose\\_var\\_priority](#) ( comb\_data \* cd, llist \* remaining\_vars, llist \* undo\_deg, int priority\_last\_n )

4.56.1.4 int [choose\\_var\\_priority\\_decided](#) ( comb\_data \* cd, llist \* remaining\_vars, llist \* undo\_deg, int priority\_last\_n, int decided, active\_tuple \* ac )

4.56.1.5 void [clear\\_comb\\_datas\\_from\\_hts](#) ( llist \* htlist )

4.56.1.6 void [destroy\\_comb](#) ( llist \* comb )

4.56.1.7 void [destroy\\_comb\\_cdata](#) ( comb\_cdata \* ccd )

4.56.1.8 void [destroy\\_comb\\_data2](#) ( comb\_data \* cd )

4.56.1.9 void [enum\\_table\\_pairs](#) ( htable \* hta0, llist\_node \* combnp, int \* varmap, comb\_data \* cd, comb\_data \* base\_cd )

4.56.1.10 int [filter\\_all\\_relations\\_with\\_all\\_solutions](#) ( comb\_data \* cd, set \* cluster\_vars, active\_tuple \* ac, double \* sol\_count )

- 4.56.1.11 `int filter_all_relations_with_all_solutions_priority ( comb_data * cd, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count )`
- 4.56.1.12 `int filter_single_relation_in_combination ( comb_data * cd, set * undo_set, light_stack * undo_stack, double * sol_count )`
- 4.56.1.13 `void find_all_supports_with_all_solutions_with_fc ( comb_data * cd, set * cluster_vars, active_tuple * ac, double * sol_count )`
- 4.56.1.14 `void find_all_supports_with_all_solutions_with_fc_priority ( comb_data * cd, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count )`
- 4.56.1.15 `void find_all_supports_with_all_solutions_with_fc_priority_next_var ( comb_data * cd, set * cluster_vars, active_tuple * ac, int priority_last_n, int next_v, double * sol_count )`
- 4.56.1.16 `int find_support_for_single_solution ( int t, comb_data * cd, set * cluster_vars, double * sol_count )`
- 4.56.1.17 `int find_support_for_single_solution_bc ( int t, comb_data * cd )`
- 4.56.1.18 `int find_support_for_single_solution_bc_rec ( int ** tuples, comb_data * cd, int level, int * tuple_positions, int * tuple_position_htid )`
- 4.56.1.19 `int find_support_for_single_solution_with_fc ( int t, comb_data * cd, int first_table, int all_solutions, llist * remaining_vars, set * cluster_vars, active_tuple * ac, double * sol_count )`
- 4.56.1.20 `int find_support_for_single_solution_with_fc_priority ( int t, comb_data * cd, int first_table, int all_solutions, llist * remaining_vars, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count )`
- 4.56.1.21 `int find_support_for_single_solution_with_fc_rec ( int rec_depth, int ** allocated_marks, int * old_marks, int ** tuples, comb_data * cd, int var, int * tuple_positions, int * tuple_position_htid, llist * remaining_vars, int all_solutions, llist *** doms, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count, llist * stack )`
- 4.56.1.22 `int find_support_for_single_solution_with_fc_rec_single_out ( int rec_depth, int ** allocated_marks, int * old_marks, int ** tuples, comb_data * cd, int var, int * tuple_positions, int * tuple_position_htid, llist * remaining_vars, int all_solutions, llist *** doms, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count, llist * stack )`
- 4.56.1.23 `llist* get_comb_from ( htable * filter, htable * given )`
- 4.56.1.24 `htable* get_ht_from_cd ( comb_data * cd, int ht_pos )`
- 4.56.1.25 `htable* get_hts_paired ( comb_data * cd, int hi, int hj, int p )`
- 4.56.1.26 `llist* get_pair_common_cols ( comb_data * cd, int hi, int hj )`
- 4.56.1.27 `int get_revision_time ( htable * ht, int t )`
- 4.56.1.28 `void get_rstat ( comb_data * cdata )`
- 4.56.1.29 `tree_map* get_tms_paired ( comb_data * cd, int hi, int hj )`
- 4.56.1.30 `comb_data* make_comb_data2 ( llist * comb, llist_node * combnp, int context, int no_tuple_positions, options category, comb_data * base_cd )`

- 4.56.1.31 `int mark_all_in_future ( int * old_marks, int mark_with, int var, llist * remaining_vars, int * current_tuple, comb_data * cd, llist * undo_dom, int * marks, llist *** doms, int rec_depth, int drop_for_all_revised )`
- 4.56.1.32 `comb_cdata* new_comb_cdata ( int context )`
- 4.56.1.33 `void queue_relation_related ( htable * ht, set * queue )`
- 4.56.1.34 `void queue_relation_related_multiqueue ( htable * ht, set * queue, set ** queues, int * queues_total_count )`
- 4.56.1.35 `void set_deg_count ( comb_data * cd, llist * remaining_vars, llist * undo_deg, int removed )`
- 4.56.1.36 `void set_dom ( comb_data * cd )`
- 4.56.1.37 `void set_revision_time ( htable * ht, int t, int revision_time, comb_data * cd )`
- 4.56.1.38 `void set_revision_time_parents ( htable * ht, int t, int revision_time, comb_data * cd, llist * stack, int ** tuples )`
- 4.56.1.39 `comb_data* setup_relation_group_for_a_comb ( llist * comb, int context, options category, comb_data * base_cd )`
- 4.56.1.40 `comb_data* setup_relation_group_for_a_comb_first ( llist * comb, int context, llist_node * firsttable, options category, comb_data * base_cd )`
- 4.56.1.41 `comb_data* setup_relation_group_for_a_comb_long_indexes ( llist * comb, options category )`
- 4.56.1.42 `int setup_relation_groups ( llist * combs, int context, options category )`
- 4.56.1.43 `int setup_relation_groups_ ( llist * combs, int context, options category, comb_data ** cdatas, llist * context_list )`
- 4.56.1.44 `int stage4_filter ( set * queue, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack, int ppr )`
- 4.56.1.45 `int stage4_filter_auto ( set * queue, set ** queues, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack, int * queues_total_count, int ppr )`
- 4.56.1.46 `int stage4_filter_auto_all_sol ( set * queue, set ** queues, int time, set * undo_set, light_stack * undo_stack, int * queues_total_count )`
- 4.56.1.47 `int stage4_filter_auto_one_sol ( set * queue, set ** queues, int time, set * undo_set, light_stack * undo_stack, int * queues_total_count )`
- 4.56.1.48 `int stage4_filter_multiqueue ( set * queue, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack, set ** queues, int * queues_total_count, int ppr )`
- 4.56.1.49 `int stage4_filter_with_all_solutions ( set * queue, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack, int * queues_total_count, int ppr )`
- 4.56.1.50 `int stage4_filter_with_no_check_insol_tuples ( set * queue, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack )`
- 4.56.1.51 `set* stage4_init_queue ( llist * constraint_graph_node_list )`
- 4.56.1.52 `void undo_dom_deg_var ( llist * undo_deg, llist * undo_dom, comb_data * cd, int var, llist * remaining_vars, llist *** doms, llist ** doms_xlevel )`
- 4.56.1.53 `int visit_rest_of_tuples ( llist * remaining_vars, int all_solutions, int next_var, comb_data * cd, int var, int * old_marks, int ** tuples, llist *** doms, int rec_depth, int * tuple_positions, llist * undo_deg, llist * undo_dom )`

## 4.56.2 Variable Documentation

4.56.2.1 `void* watch = NULL`

## 4.57 rs2.h File Reference

```
#include "query_graph.h"
#include "llist.h"
#include "hashtable.h"
#include "set.h"
#include "tree_map.h"
#include "rs.h"
#include "active_tuple.h"
```

### Classes

- struct [table\\_map](#)
- struct [comb\\_cdata](#)

### Typedefs

- typedef struct [table\\_map](#) [table\\_map](#)
- typedef struct [comb\\_cdata](#) [comb\\_cdata](#)

### Functions

- [htable](#) \* [get\\_hts\\_paired](#) ([comb\\_data](#) \*cd, int hi, int hj, int p)
- [tree\\_map](#) \* [get\\_tms\\_paired](#) ([comb\\_data](#) \*cd, int hi, int hj)
- [llist](#) \* [get\\_pair\\_common\\_cols](#) ([comb\\_data](#) \*cd, int hi, int hj)
- [comb\\_data](#) \* [make\\_comb\\_data2](#) ([llist](#) \*comb, [llist\\_node](#) \*combnp, int context, int no\_tuple\_positions, [options](#) category, [comb\\_data](#) \*base\_cd)
- void [destroy\\_comb\\_data2](#) ([comb\\_data](#) \*cd)
- void [filter\\_relations\\_rs](#) ([llist](#) \*queue)
- void [build\\_joining\\_order](#) ([htable](#) \*hta, [llist\\_node](#) \*combnp, int \*vmap, [llist](#) \*hts\_ordered, [llist](#) \*all\_com\_cols)
- void [build\\_ind\\_trees](#) ([comb\\_data](#) \*cd, [llist\\_node](#) \*combnp, int \*vmap, [comb\\_data](#) \*base\_cd)
- int [find\\_support\\_for\\_single\\_solution](#) (int t, [comb\\_data](#) \*cd, [set](#) \*cluster\_vars, double \*sol\_count)
- int [find\\_support\\_for\\_single\\_solution\\_rec](#) (int \*\*tuples, [comb\\_data](#) \*cd, int level, int \*tuple\_positions, int \*tuple\_position\_htid)
- int [setup\\_relation\\_groups](#) ([llist](#) \*combs, int context, [options](#) category)
- int [setup\\_relation\\_groups\\_](#) ([llist](#) \*combs, int context, [options](#) category, [comb\\_data](#) \*\*cdatas, [llist](#) \*context\_list)
- [set](#) \* [stage4\\_init\\_queue](#) ([llist](#) \*constraint\_graph\_node\_list)
- int [stage4\\_filter\\_auto](#) ([set](#) \*queue, [set](#) \*\*queues, int time, [set](#) \*undo\_set, [set](#) \*cluster\_vars, [light\\_stack](#) \*undo\_stack, int \*queues\_total\_count, int ppr)
- int [stage4\\_filter\\_auto\\_one\\_sol](#) ([set](#) \*queue, [set](#) \*\*queues, int time, [set](#) \*undo\_set, [light\\_stack](#) \*undo\_stack, int \*queues\_total\_count)
- int [stage4\\_filter\\_auto\\_all\\_sol](#) ([set](#) \*queue, [set](#) \*\*queues, int time, [set](#) \*undo\_set, [light\\_stack](#) \*undo\_stack, int \*queues\_total\_count)
- int [stage4\\_filter](#) ([set](#) \*queue, int time, [set](#) \*undo\_set, [set](#) \*cluster\_vars, [light\\_stack](#) \*undo\_stack, int ppr)
- int [stage4\\_filter\\_multiqueue](#) ([set](#) \*queue, int time, [set](#) \*undo\_set, [set](#) \*cluster\_vars, [light\\_stack](#) \*undo\_stack, [set](#) \*\*queues, int \*queues\_total\_count, int ppr)
- void [queue\\_related\\_relations](#) ([set](#) \*queue, [htable](#) \*ht)



- support \* [new\\_support](#) (comb\_data \*comb, int tuple)
- void [destroy\\_support](#) (support \*s)
- update\_support \* [new\\_update\\_support](#) (comb\_data \*comb)
- void [queue\\_supported\\_tuples](#) (llist \*supports, set \*updated\_combinations)
- void [remove\\_changed\\_supports](#) (set \*updated\_combinations)
- set \* [new\\_updated\\_combinations](#) (int combinations)
- void [clear\\_support\\_tuple\\_sets](#) (set \*updated\_combinations)
- void [enum\\_table\\_pairs](#) (htable \*hta0, llist\_node \*combnp, int \*vmap, comb\_data \*cd, comb\_data \*base\_cd)
- int [stage4\\_filter\\_with\\_no\\_check\\_insol\\_tuples](#) (set \*queue, int time, set \*undo\_set, set \*cluster\_vars, light\_stack \*undo\_stack)
- int [find\\_support\\_for\\_single\\_solution\\_bc](#) (int t, comb\_data \*cd)
- int [find\\_support\\_for\\_single\\_solution\\_bc\\_rec](#) (int \*\*tuples, comb\_data \*cd, int level, int \*tuple\_positions, int \*tuple\_position\_htid)
- int [find\\_support\\_for\\_single\\_solution\\_with\\_fc](#) (int t, comb\_data \*cd, int first\_table, int all\_solutions, llist \*remaining\_vars, set \*cluster\_vars, active\_tuple \*ac, double \*sol\_count)
- int [mark\\_all\\_in\\_future](#) (int \*old\_marks, int mark\_with, int var, llist \*remaining\_vars, int \*current\_tuple, comb\_data \*cd, llist \*undo\_dom, int \*marks, llist \*\*\*doms, int rec\_depth, int drop\_for\_all\_revised)
- int [find\\_support\\_for\\_single\\_solution\\_with\\_fc\\_rec\\_single\\_out](#) (int rec\_depth, int \*\*allocated\_marks, int \*old\_marks, int \*\*tuples, comb\_data \*cd, int var, int \*tuple\_positions, int \*tuple\_position\_htid, llist \*remaining\_vars, int all\_solutions, llist \*\*\*doms, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, double \*sol\_count, llist \*stack)
- int [find\\_support\\_for\\_single\\_solution\\_with\\_fc\\_rec](#) (int rec\_depth, int \*\*allocated\_marks, int \*old\_marks, int \*\*tuples, comb\_data \*cd, int var, int \*tuple\_positions, int \*tuple\_position\_htid, llist \*remaining\_vars, int all\_solutions, llist \*\*\*doms, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, double \*sol\_count, llist \*stack)
- int [stage4\\_filter\\_with\\_all\\_solutions](#) (set \*queue, int time, set \*undo\_set, set \*cluster\_vars, light\_stack \*undo\_stack, int \*queues\_total\_count, int ppr)
- void [find\\_all\\_supports\\_with\\_all\\_solutions\\_with\\_fc](#) (comb\_data \*cd, set \*cluster\_varst, active\_tuple \*ac, double \*sol\_count)
- void [set\\_dom](#) (comb\_data \*cd)
- int [choose\\_var](#) (comb\_data \*cd, llist \*remaining\_vars, llist \*undo\_deg)
- int [choose\\_var\\_priority](#) (comb\_data \*cd, llist \*remaining\_vars, llist \*undo\_deg, int priority\_first\_n)
- int [choose\\_var\\_priority\\_decided](#) (comb\_data \*cd, llist \*remaining\_vars, llist \*undo\_deg, int priority\_last\_n, int decided, active\_tuple \*ac)
- void [undo\\_dom\\_deg\\_var](#) (llist \*undo\_deg, llist \*undo\_dom, comb\_data \*cd, int var, llist \*remaining\_vars, llist \*\*\*doms, llist \*\*doms\_xlevel)
- void [set\\_deg\\_count](#) (comb\_data \*cd, llist \*remaining\_vars, llist \*undo\_deg, int removed)
- htable \* [get\\_ht\\_from\\_cd](#) (comb\_data \*cd, int ht\_pos)
- void [set\\_revision\\_time](#) (htable \*ht, int t, int revision\_time, comb\_data \*cd)
- void [set\\_revision\\_time\\_parents](#) (htable \*ht, int t, int revision\_time, comb\_data \*cd, llist \*stack, int \*\*tuples)
- int [get\\_revision\\_time](#) (htable \*ht, int t)
- int [visit\\_rest\\_of\\_tuples](#) (llist \*remaining\_vars, int all\_solutions, int next\_var, comb\_data \*cd, int var, int \*old\_marks, int \*\*tuples, llist \*\*\*doms, int rec\_depth, int \*tuple\_positions, llist \*undo\_deg, llist \*undo\_dom)
- int [filter\\_all\\_relations\\_with\\_all\\_solutions](#) (comb\_data \*cd, set \*cluster\_vars, active\_tuple \*ac, double \*sol\_count)
- int [filter\\_single\\_relation\\_in\\_combination](#) (comb\_data \*cd, set \*undo\_set, light\_stack \*undo\_stack, double \*sol\_count)
- comb\_data \* [setup\\_relation\\_group\\_for\\_a\\_comb\\_first](#) (llist \*comb, int context, llist\_node \*firsttable, options category, comb\_data \*base\_cd)
- comb\_data \* [setup\\_relation\\_group\\_for\\_a\\_comb](#) (llist \*comb, int context, options category, comb\_data \*base\_cd)
- comb\_data \* [setup\\_relation\\_group\\_for\\_a\\_comb\\_long\\_indexes](#) (llist \*comb, options category)
- int [filter\\_all\\_relations\\_with\\_all\\_solutions\\_priority](#) (comb\_data \*cd, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, double \*sol\_count)
- int [find\\_support\\_for\\_single\\_solution\\_with\\_fc\\_priority](#) (int t, comb\_data \*cd, int first\_table, int all\_solutions, llist \*remaining\_vars, set \*cluster\_vars, active\_tuple \*ac, int priority\_last\_n, double \*sol\_count)

- void `find_all_supports_with_all_solutions_with_fc_priority` (`comb_data *cd`, `set *cluster_vars`, `active_tuple *ac`, `int priority_last_n`, `double *sol_count`)
- void `find_all_supports_with_all_solutions_with_fc_priority_next_var` (`comb_data *cd`, `set *cluster_vars`, `active_tuple *ac`, `int priority_last_n`, `int next_v`, `double *sol_count`)
- void `queue_relation_related` (`htable *ht`, `set *queue`)
- void `queue_relation_related_multiqueue` (`htable *ht`, `set *queue`, `set **queues`, `int *queues_total_count`)
- `comb_cdata * new_comb_cdata` (`int context`)
- void `destroy_comb_cdata` (`comb_cdata *ccd`)
- `llist * get_comb_from` (`htable *filter`, `htable *given`)
- void `destroy_comb` (`llist *comb`)
- void `get_rstat` (`comb_data *cdata`)
- void `clear_comb_datas_from_hts` (`llist *htlist`)

### 4.57.1 Detailed Description

Provides helper methods for initializing the data-structures for the PerTuple and AllSol algorithms.

### 4.57.2 Typedef Documentation

4.57.2.1 `typedef struct comb_cdata comb_cdata`

4.57.2.2 `typedef struct table_map table_map`

### 4.57.3 Function Documentation

4.57.3.1 void `build_ind_trees` ( `comb_data * cd`, `llist_node * combnp`, `int * vmap`, `comb_data * base_cd` )

4.57.3.2 void `build_joining_order` ( `htable * hta`, `llist_node * combnp`, `int * vmap`, `llist * hts_ordered`, `llist * all_com_cols` )

4.57.3.3 int `choose_var` ( `comb_data * cd`, `llist * remaining_vars`, `llist * undo_deg` )

4.57.3.4 int `choose_var_priority` ( `comb_data * cd`, `llist * remaining_vars`, `llist * undo_deg`, `int priority_first_n` )

4.57.3.5 int `choose_var_priority_decided` ( `comb_data * cd`, `llist * remaining_vars`, `llist * undo_deg`, `int priority_last_n`, `int decided`, `active_tuple * ac` )

4.57.3.6 void `clear_comb_datas_from_hts` ( `llist * htlist` )

4.57.3.7 void `clear_support_tuple_sets` ( `set * updated_combinations` )

4.57.3.8 void `destroy_comb` ( `llist * comb` )

4.57.3.9 void `destroy_comb_cdata` ( `comb_cdata * ccd` )

4.57.3.10 void `destroy_comb_data2` ( `comb_data * cd` )

4.57.3.11 void `destroy_support` ( `support * s` )

4.57.3.12 void `enum_table_pairs` ( `htable * hta0`, `llist_node * combnp`, `int * vmap`, `comb_data * cd`, `comb_data * base_cd` )

4.57.3.13 int `filter_all_relations_with_all_solutions` ( `comb_data * cd`, `set * cluster_vars`, `active_tuple * ac`, `double * sol_count` )

- 4.57.3.14 `int filter_all_relations_with_all_solutions_priority ( comb_data * cd, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count )`
- 4.57.3.15 `void filter_relations_rs ( llist * queue )`
- 4.57.3.16 `int filter_single_relation_in_combination ( comb_data * cd, set * undo_set, light_stack * undo_stack, double * sol_count )`
- 4.57.3.17 `void find_all_supports_with_all_solutions_with_fc ( comb_data * cd, set * cluster_varst, active_tuple * ac, double * sol_count )`
- 4.57.3.18 `void find_all_supports_with_all_solutions_with_fc_priority ( comb_data * cd, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count )`
- 4.57.3.19 `void find_all_supports_with_all_solutions_with_fc_priority_next_var ( comb_data * cd, set * cluster_vars, active_tuple * ac, int priority_last_n, int next_v, double * sol_count )`
- 4.57.3.20 `int find_support_for_single_solution ( int t, comb_data * cd, set * cluster_vars, double * sol_count )`
- 4.57.3.21 `int find_support_for_single_solution_bc ( int t, comb_data * cd )`
- 4.57.3.22 `int find_support_for_single_solution_bc_rec ( int ** tuples, comb_data * cd, int level, int * tuple_positions, int * tuple_position_htid )`
- 4.57.3.23 `int find_support_for_single_solution_rec ( int ** tuples, comb_data * cd, int level, int * tuple_positions, int * tuple_position_htid )`
- 4.57.3.24 `int find_support_for_single_solution_with_fc ( int t, comb_data * cd, int first_table, int all_solutions, llist * remaining_vars, set * cluster_vars, active_tuple * ac, double * sol_count )`
- 4.57.3.25 `int find_support_for_single_solution_with_fc_priority ( int t, comb_data * cd, int first_table, int all_solutions, llist * remaining_vars, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count )`
- 4.57.3.26 `int find_support_for_single_solution_with_fc_rec ( int rec_depth, int ** allocated_marks, int * old_marks, int ** tuples, comb_data * cd, int var, int * tuple_positions, int * tuple_position_htid, llist * remaining_vars, int all_solutions, llist *** doms, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count, llist * stack )`
- 4.57.3.27 `int find_support_for_single_solution_with_fc_rec_single_out ( int rec_depth, int ** allocated_marks, int * old_marks, int ** tuples, comb_data * cd, int var, int * tuple_positions, int * tuple_position_htid, llist * remaining_vars, int all_solutions, llist *** doms, set * cluster_vars, active_tuple * ac, int priority_last_n, double * sol_count, llist * stack )`
- 4.57.3.28 `llist* get_comb_from ( htable * filter, htable * given )`
- 4.57.3.29 `htable* get_ht_from_cd ( comb_data * cd, int ht_pos )`
- 4.57.3.30 `htable* get_hts_paired ( comb_data * cd, int hi, int hj, int p )`
- 4.57.3.31 `llist* get_pair_common_cols ( comb_data * cd, int hi, int hj )`
- 4.57.3.32 `int get_revision_time ( htable * ht, int t )`
- 4.57.3.33 `void get_rstat ( comb_data * cdata )`
- 4.57.3.34 `tree_map* get_tms_paired ( comb_data * cd, int hi, int hj )`

- 4.57.3.35 `comb_data* make_comb_data2 ( llist * comb, llist_node * combnp, int context, int no_tuple_positions, options category, comb_data * base_cd )`
- 4.57.3.36 `int mark_all_in_future ( int * old_marks, int mark_with, int var, llist * remaining_vars, int * current_tuple, comb_data * cd, llist * undo_dom, int * marks, llist *** doms, int rec_depth, int drop_for_all_revised )`
- 4.57.3.37 `comb_cdata* new_comb_cdata ( int context )`
- 4.57.3.38 `support* new_support ( comb_data * comb, int tuple )`
- 4.57.3.39 `update_support* new_update_support ( comb_data * comb )`
- 4.57.3.40 `set* new_updated_combinations ( int combinations )`
- 4.57.3.41 `void queue_related_relations ( set * queue, htable * ht )`
- 4.57.3.42 `void queue_relation_related ( htable * ht, set * queue )`
- 4.57.3.43 `void queue_relation_related_multiqueue ( htable * ht, set * queue, set ** queues, int * queues_total_count )`
- 4.57.3.44 `void queue_supported_tuples ( llist * supports, set * updated_combinations )`
- 4.57.3.45 `void remove_changed_supports ( set * updated_combinations )`
- 4.57.3.46 `void set_deg_count ( comb_data * cd, llist * remaining_vars, llist * undo_deg, int removed )`
- 4.57.3.47 `void set_dom ( comb_data * cd )`
- 4.57.3.48 `void set_revision_time ( htable * ht, int t, int revision_time, comb_data * cd )`
- 4.57.3.49 `void set_revision_time_parents ( htable * ht, int t, int revision_time, comb_data * cd, llist * stack, int ** tuples )`
- 4.57.3.50 `comb_data* setup_relation_group_for_a_comb ( llist * comb, int context, options category, comb_data * base_cd )`
- 4.57.3.51 `comb_data* setup_relation_group_for_a_comb_first ( llist * comb, int context, llist_node * firstable, options category, comb_data * base_cd )`
- 4.57.3.52 `comb_data* setup_relation_group_for_a_comb_long_indexes ( llist * comb, options category )`
- 4.57.3.53 `int setup_relation_groups ( llist * combs, int context, options category )`
- 4.57.3.54 `int setup_relation_groups_ ( llist * combs, int context, options category, comb_data ** cdatas, llist * context_list )`
- 4.57.3.55 `int stage4_filter ( set * queue, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack, int ppr )`
- 4.57.3.56 `int stage4_filter_auto ( set * queue, set ** queues, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack, int * queues_total_count, int ppr )`
- 4.57.3.57 `int stage4_filter_auto_all_sol ( set * queue, set ** queues, int time, set * undo_set, light_stack * undo_stack, int * queues_total_count )`
- 4.57.3.58 `int stage4_filter_auto_one_sol ( set * queue, set ** queues, int time, set * undo_set, light_stack * undo_stack, int * queues_total_count )`

- 4.57.3.59 `int stage4_filter_multiqueue ( set * queue, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack, set ** queues, int * queues_total_count, int ppr )`
- 4.57.3.60 `int stage4_filter_with_all_solutions ( set * queue, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack, int * queues_total_count, int ppr )`
- 4.57.3.61 `int stage4_filter_with_no_check_insol_tuples ( set * queue, int time, set * undo_set, set * cluster_vars, light_stack * undo_stack )`
- 4.57.3.62 `set* stage4_init_queue ( llist * constraint_graph_node_list )`
- 4.57.3.63 `void undo_dom_deg_var ( llist * undo_deg, llist * undo_dom, comb_data * cd, int var, llist * remaining_vars, llist ** doms, llist ** doms_xlevel )`
- 4.57.3.64 `int visit_rest_of_tuples ( llist * remaining_vars, int all_solutions, int next_var, comb_data * cd, int var, int * old_marks, int ** tuples, llist *** doms, int rec_depth, int * tuple_positions, llist * undo_deg, llist * undo_dom )`

## 4.58 rs2tree.c File Reference

```
#include "rs2tree.h"
#include <math.h>
#include "combinations_wrapper.h"
#include "rs2.h"
#include "rsprocedures.h"
#include "tree.h"
#include "btdallsol.h"
#include "domains.h"
#include "hashtable_search.h"
#include "decomposition_functions.h"
#include "decomposition.h"
#include "rr.h"
#include "tables.h"
#include "gbtd.h"
#include "septables.h"
#include "sort.h"
```

### Functions

- `int rs2_all_sols_bt_pre_processing (decomposed_tree *treen, rs2tree_configuration *config, int ppr)`
- `void pre_feature_extract (rb_tree **stats, decomposed_tree *treen)`
- `void post_feature_extract (rb_tree *stats, decomposed_tree *treen, int number_of_clusters)`
- `void rs2_all_sols_bt (decomposed_tree *treen)`
- `int process_tree_combinations (decomposed_tree *treen, set *undo_set, light_stack *undo_stack, set *queue, int ppr)`
- `int process_tree_combinations2 (decomposed_tree *treen, set *undo_set, light_stack *undo_stack, int *repeat, int isroot, set **queues, int queues_count, int onelevel, int ppr)`
- `int process_tree_combinations3 (decomposed_tree *treen, set *undo_set, light_stack *undo_stack, int *repeat, int isroot, set **queues, int queues_count, int onelevel)`
- `rs2tree_stats * get_make_stats (rb_tree **stats, decomposed_tree *treen)`
- `void init_update_processed_info (rb_tree **stats, decomposed_tree *treen, int init, int update)`
- `void init_processed_info (rb_tree **stats, decomposed_tree *treen)`
- `void update_processed_info (rb_tree **stats, decomposed_tree *treen)`
- `void rs2t_set_heap_pos (int val, void *body)`
- `void rs2t_set_heap_value (int val, void *body)`

- int [rs2t\\_get\\_heap\\_value](#) (void \*body)
- int [rs2t\\_get\\_id](#) (void \*body)
- void [check\\_this](#) (set \*\*queues, int count, int sum)
- [decomposed\\_tree](#) \* [fringe\\_extract\\_top](#) (heap \*h)
- int [process\\_tree\\_combinations5](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations52](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, [light\\_stack](#) \*value\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations522](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, [light\\_stack](#) \*value\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_this\\_cluster](#) ([decomposed\\_tree](#) \*current, set \*undo\_set, [light\\_stack](#) \*undo\_stack, [light\\_stack](#) \*value\_stack, set \*\*queues, [decomposed\\_tree](#) \*root, int \*aqueues\_total\_count, int ppr)
- int [process\\_tree\\_combinations53](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations54](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations6](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations7](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations8](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, [light\\_stack](#) \*value\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations9](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations4](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, [light\\_stack](#) \*value\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations4\\_](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, variable \*var, int ppr)
- int [process\\_tree\\_combinations4\\_filter\\_up\\_rec](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, int ppr)
- int [process\\_tree\\_combinations4\\_filter\\_down\\_rec](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, int ppr)
- int [process\\_tree\\_combinations4\\_filter](#) ([decomposed\\_tree](#) \*treen, set \*undo\_set, [light\\_stack](#) \*undo\_stack, set \*\*queues, int queues\_count, int ppr)
- void [undo\\_tuple\\_deletions\\_for\\_rs2tree](#) (set \*undo\_set, [light\\_stack](#) \*undo\_stack)
- void [assign\\_table\\_context](#) (l~~l~~ist \*usedhts, [decomposed\\_tree](#) \*treen)
- [decomposed\\_tree](#) \* [find\\_top\\_root\\_with\\_scope](#) (int \*vars, [decomposed\\_tree](#) \*root)
- int [startBTD](#) ([decomposed\\_tree](#) \*treen, long double \*total\_sols\_ret, set \*queue, set \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, int print\_sol)
- variable \* [un\\_assign\\_variable](#) ([light\\_stack](#) \*assigned\_var\_stack, [light\\_stack](#) \*undo\_set\_stack, [light\\_stack](#) \*undo\_tuple\_stack, [light\\_stack](#) \*undo\_value\_stack)
- void [undo\\_value\\_deletions](#) ([light\\_stack](#) \*value\_stack)
- int [assign\\_variable](#) (variable \*var)
- variable \* [choose\\_var\\_from\\_treen](#) ([decomposed\\_tree](#) \*treen)
- set \* [set\\_variable\\_htables2](#) ([decomposed\\_tree](#) \*\*c\_array, int clusters)
- void [set\\_queues\\_treenodes2\\_singleq](#) ([decomposed\\_tree](#) \*\*treens, int clusters, l~~l~~ist \*tables, set \*\*queues)
- void [set\\_queues\\_treenodes2](#) ([decomposed\\_tree](#) \*\*treens, int clusters, l~~l~~ist \*tables, set \*\*queues)
- void [set\\_single\\_graph\\_combinations\\_to\\_treenodes](#) ([decomposed\\_tree](#) \*treen, l~~l~~ist \*combs, [constraint\\_graph](#) \*cg)
- void [single\\_graph\\_combinations\\_with\\_tables](#) (l~~l~~ist \*tables, l~~l~~ist \*\*ret\_combs, [constraint\\_graph](#) \*\*ret\_cg)
- int [filter\\_table\\_with\\_chosen\\_value](#) (htable \*ht, [light\\_stack](#) \*undo\_stack, set \*undo\_set, variable \*var)
- void [reset\\_variable\\_domain](#) (variable \*var)
- void [reset\\_variable\\_domain2](#) (variable \*var)
- void [record\\_good\\_nogood](#) (int good, long double weight, [decomposed\\_tree](#) \*child)
- void [set\\_good\\_nogood\\_trees](#) ([decomposed\\_tree](#) \*treen)
- void [rs2tree\\_filter\\_domains](#) (variables \*vars, [light\\_stack](#) \*value\_stack)

- void `reset_relation_queues` (`set **queues`, `int count`)
- void `set_cluster_constraints_to_projected_reols_on_c_vars` (`decomposed_tree *treen`)
- void `set_default_context_to_tables` (`decomposed_tree *treen`)
- void \* `get_payload` (`int param`)
- void `traverse_treemap_set_collect_counts` (`tree_map *tm`, `int width`, `llist *result`, `double norm_factor`)
- void `report_tree` (`decomposed_tree *treen`, `rb_tree **stats`, `int initial`)
- double \* `avgStdev2` (`rs2tree_stats *st`, `double **array`)
- double \* `avgStdev` (`decomposed_tree *treen`, `double **array`)
- `rs2tree_stats * new_rs2tree_stats` (`void`)
- void `extract_feature` (`rs2tree_stats *rs_stat`, `int ppr`)
- void `add_times_to_stats` (`rb_tree *stats`)
- void `print_rs2tree_stats` (`rb_tree *stats`, `int count`, `int thiscluster`)
- int `start_w_b_BT` (`decomposed_tree *treen`, `long double *total_sols_ret`, `set *queue`, `set **queues`, `int queues_count`, `decomposed_tree **treens`, `int all_solutions`)
- int `rs2tree_filter_relations_given_domains` (`llist *separatorvars`, `light_stack *undo_stack`, `light_stack *value_stack`, `set *undo_set`, `set *queue`, `set **queues`, `int *queues_total_count`, `llist *propagator_vars`)
- void `rs2tree_filter_domains_given_relations` (`llist *constraints`, `llist *separatorvars`, `light_stack *value_stack`)
- void `rs2tree_filter_domains_given_relation` (`htable *rel`, `light_stack *value_stack`, `llist *propagator_vars`)
- int `process_geneate_bolstering_bin_cliquequeue_relations` (`decomposed_tree *root`)

## 4.58.1 Function Documentation

4.58.1.1 void `add_times_to_stats` ( `rb_tree * stats` )

4.58.1.2 void `assign_table_context` ( `llist * usedhts`, `decomposed_tree * treen` )

4.58.1.3 int `assign_variable` ( `variable * var` )

4.58.1.4 double\* `avgStdev` ( `decomposed_tree * treen`, `double ** array` )

4.58.1.5 double\* `avgStdev2` ( `rs2tree_stats * st`, `double ** array` )

4.58.1.6 void `check_this` ( `set ** queues`, `int count`, `int sum` )

4.58.1.7 variable\* `choose_var_from_treen` ( `decomposed_tree * treen` )

4.58.1.8 void `extract_feature` ( `rs2tree_stats * rs_stat`, `int ppr` )

4.58.1.9 int `filter_table_with_chosen_value` ( `htable * ht`, `light_stack * undo_stack`, `set * undo_set`, `variable * var` )

4.58.1.10 `decomposed_tree * find_top_root_with_scope` ( `int * vars`, `decomposed_tree * root` )

4.58.1.11 `decomposed_tree* fringe_extract_top` ( `heap * h` )

4.58.1.12 `rs2tree_stats* get_make_stats` ( `rb_tree ** stats`, `decomposed_tree * treen` )

4.58.1.13 void\* `get_payload` ( `int param` )

4.58.1.14 void `init_processed_info` ( `rb_tree ** stats`, `decomposed_tree * treen` )

4.58.1.15 void `init_update_processed_info` ( `rb_tree ** stats`, `decomposed_tree * treen`, `int init`, `int update` )

4.58.1.16 `rs2tree_stats* new_rs2tree_stats` ( `void` )

4.58.1.17 void `post_feature_extract` ( `rb_tree * stats`, `decomposed_tree * treen`, `int number_of_clusters` )

- 4.58.1.18 void pre\_feature\_extract ( rb\_tree \*\* stats, decomposed\_tree \* treeen )
- 4.58.1.19 void print\_rs2tree\_stats ( rb\_tree \* stats, int count, int thiscluster )
- 4.58.1.20 int process\_geneate\_bolstering\_bin\_cliquote\_relations ( decomposed\_tree \* root )
- 4.58.1.21 int process\_this\_cluster ( decomposed\_tree \* current, set \* undo\_set, light\_stack \* undo\_stack, light\_stack \* value\_stack, set \*\* queues, decomposed\_tree \* root, int \* aqueues\_total\_count, int ppr )
- 4.58.1.22 int process\_tree\_combinations ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \* queue, int ppr )
- 4.58.1.23 int process\_tree\_combinations2 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, int \* repeat, int isroot, set \*\* queues, int queues\_count, int onelevel, int ppr )
- 4.58.1.24 int process\_tree\_combinations3 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, int \* repeat, int isroot, set \*\* queues, int queues\_count, int onelevel )
- 4.58.1.25 int process\_tree\_combinations4 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, light\_stack \* value\_stack, set \*\* queues, int queues\_count, decomposed\_tree \*\* treens, variable \* var, int ppr )
- 4.58.1.26 int process\_tree\_combinations4\_ ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \*\* queues, int queues\_count, decomposed\_tree \*\* treens, variable \* var, int ppr )
- 4.58.1.27 int process\_tree\_combinations4\_filter ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \*\* queues, int queues\_count, int ppr )
- 4.58.1.28 int process\_tree\_combinations4\_filter\_down\_rec ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \*\* queues, int queues\_count, int ppr )
- 4.58.1.29 int process\_tree\_combinations4\_filter\_up\_rec ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \*\* queues, int queues\_count, int ppr )
- 4.58.1.30 int process\_tree\_combinations5 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \*\* queues, int queues\_count, decomposed\_tree \*\* treens, variable \* var, int ppr )
- 4.58.1.31 int process\_tree\_combinations52 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, light\_stack \* value\_stack, set \*\* queues, int queues\_count, decomposed\_tree \*\* treens, variable \* var, int ppr )
- 4.58.1.32 int process\_tree\_combinations522 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, light\_stack \* value\_stack, set \*\* queues, int queues\_count, decomposed\_tree \*\* treens, variable \* var, int ppr )
- 4.58.1.33 int process\_tree\_combinations53 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \*\* queues, int queues\_count, decomposed\_tree \*\* treens, variable \* var, int ppr )
- 4.58.1.34 int process\_tree\_combinations54 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \*\* queues, int queues\_count, decomposed\_tree \*\* treens, variable \* var, int ppr )
- 4.58.1.35 int process\_tree\_combinations6 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \*\* queues, int queues\_count, decomposed\_tree \*\* treens, variable \* var, int ppr )
- 4.58.1.36 int process\_tree\_combinations7 ( decomposed\_tree \* treeen, set \* undo\_set, light\_stack \* undo\_stack, set \*\* queues, int queues\_count, decomposed\_tree \*\* treens, variable \* var, int ppr )



- 4.58.1.37 `int process_tree_combinations8 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, light_stack * value_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.58.1.38 `int process_tree_combinations9 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.58.1.39 `void record_good_nogood ( int good, long double weight, decomposed_tree * child )`
- 4.58.1.40 `void report_tree ( decomposed_tree * treen, rb_tree ** stats, int initial )`
- for ( np = *treen*->children->head; np != NULL; np = np->next ) { `decomposed_tree` \* *child* = (decomposed\_tree\*)np->body; report\_tree ( *child*, *stats*, *initial*, *queues* ); }
- 4.58.1.41 `void reset_relation_queues ( set ** queues, int count )`
- 4.58.1.42 `void reset_variable_domain ( variable * var )`
- 4.58.1.43 `void reset_variable_domain2 ( variable * var )`
- 4.58.1.44 `void rs2_all_sols_bt ( decomposed_tree * treen )`
- 4.58.1.45 `int rs2_all_sols_bt_pre_processing ( decomposed_tree * treen, rs2tree_configuration * config, int ppr )`
- 4.58.1.46 `int rs2t_get_heap_value ( void * body )`
- 4.58.1.47 `int rs2t_get_id ( void * body )`
- 4.58.1.48 `void rs2t_set_heap_pos ( int val, void * body )`
- 4.58.1.49 `void rs2t_set_heap_value ( int val, void * body )`
- 4.58.1.50 `void rs2tree_filter_domains ( variables * vars, light_stack * value_stack )`
- 4.58.1.51 `void rs2tree_filter_domains_given_relation (htable * rel, light_stack * value_stack, llist * propagator_vars )`
- 4.58.1.52 `void rs2tree_filter_domains_given_relations ( llist * constraints, llist * separatorvars, light_stack * value_stack )`
- 4.58.1.53 `int rs2tree_filter_relations_given_domains ( llist * separatorvars, light_stack * undo_stack, light_stack * value_stack, set * undo_set, set * queue, set ** queues, int * queues_total_count, llist * propagator_vars )`
- 4.58.1.54 `void set_cluster_constraints_to_projected_reols_on_c_vars ( decomposed_tree * treen )`
- 4.58.1.55 `void set_default_context_to_tables ( decomposed_tree * treen )`
- 4.58.1.56 `void set_good_nogood_trees ( decomposed_tree * treen )`
- 4.58.1.57 `void set_queues_treenodes2 ( decomposed_tree ** treens, int clusters, llist * tables, set ** queues )`
- 4.58.1.58 `void set_queues_treenodes2_singleq ( decomposed_tree ** treens, int clusters, llist * tables, set ** queues )`
- 4.58.1.59 `void set_single_graph_combinations_to_treenodes ( decomposed_tree * treen, llist * combs, constraint_graph * cg )`
- 4.58.1.60 `set* set_variable_htables2 ( decomposed_tree ** c_array, int clusters )`

- 4.58.1.61 `void single_graph_combinations_with_tables ( llist * tables, llist ** ret_combs, constraint_graph ** ret_cg )`
- 4.58.1.62 `int start_w_b_BTD ( decomposed_tree * treen, long double * total_sols_ret, set * queue, set ** queues, int queues_count, decomposed_tree ** treens, int all_solutions )`
- 4.58.1.63 `int startBTD ( decomposed_tree * treen, long double * total_sols_ret, set * queue, set ** queues, int queues_count, decomposed_tree ** treens, int print_sol )`
- 4.58.1.64 `void traverse_treemap_set_collect_counts ( tree_map * tm, int width, llist * result, double norm_factor )`
- 4.58.1.65 `variable* un_assign_variable ( light_stack * assigned_var_stack, light_stack * undo_set_stack, light_stack * undo_tuple_stack, light_stack * undo_value_stack )`
- 4.58.1.66 `void undo_tuple_deletions_for_rs2tree ( set * undo_set, light_stack * undo_stack )`
- 4.58.1.67 `void undo_value_deletions ( light_stack * value_stack )`
- 4.58.1.68 `void update_processed_info ( rb_tree ** stats, decomposed_tree * treen )`

## 4.59 rs2tree.h File Reference

```
#include "decomposition.h"
#include "llist.h"
#include "query_graph.h"
```

### Classes

- struct [rs2tree\\_configuration](#)
- struct [rs2tree\\_stats](#)

### Typedefs

- typedef struct [rs2tree\\_configuration](#) [rs2tree\\_configuration](#)
- typedef struct [rs2tree\\_stats](#) [rs2tree\\_stats](#)

### Functions

- [rs2tree\\_stats](#) \* [new\\_rs2tree\\_stats](#) (void)
- void [set\\_combinations](#) ([decomposed\\_tree](#) \*treen, [set](#) \*\*queues)
- int [set\\_combinations2](#) ([decomposed\\_tree](#) \*treen, [set](#) \*\*queues)
- int [process\\_tree\\_combinations](#) ([decomposed\\_tree](#) \*treen, [set](#) \*undo\_set, [light\\_stack](#) \*undo\_stack, [set](#) \*queue, int ppr)
- int [process\\_tree\\_combinations2](#) ([decomposed\\_tree](#) \*treen, [set](#) \*undo\_set, [light\\_stack](#) \*undo\_stack, int \*repeat, int isroot, [set](#) \*\*queues, int queues\_count, int onelevel, int ppr)
- void [undo\\_tuple\\_deletions\\_for\\_rs2tree](#) ([set](#) \*undo\_set, [light\\_stack](#) \*undo\_stack)
- int [startBTD](#) ([decomposed\\_tree](#) \*treen, long double \*total\_sols\_ret, [set](#) \*queue, [set](#) \*\*queues, int queues\_count, [decomposed\\_tree](#) \*\*treens, int print\_sol)
- [variable](#) \* [choose\\_var\\_from\\_treen](#) ([decomposed\\_tree](#) \*treen)
- [variable](#) \* [un\\_assign\\_variable](#) ([light\\_stack](#) \*assigned\_var\_stack, [light\\_stack](#) \*undo\_set\_stack, [light\\_stack](#) \*undo\_tuple\_stack, [light\\_stack](#) \*undo\_value\_stack)
- int [assign\\_variable](#) ([variable](#) \*var)
- int [filter\\_table\\_with\\_chosen\\_value](#) ([htable](#) \*ht, [light\\_stack](#) \*undo\_stack, [set](#) \*undo\_set, [variable](#) \*var)

- void `undo_value_deletions` (`light_stack *undo_value_stack`)
- int `rs2_all_sols_bt_pre_processing` (`decomposed_tree *treen`, `rs2tree_configuration *config`, int ppr)
- void `rs2_all_sols_bt` (`decomposed_tree *treen`)
- void `reset_variable_domain` (`variable *var`)
- void `reset_variable_domain2` (`variable *var`)
- void `single_graph_combinations_with_tables` (`llist *tables`, `llist **ret_combs`, `constraint_graph **ret_cg`)
- void `set_single_graph_combinations_to_treenodes` (`decomposed_tree *treen`, `llist *combs`, `constraint_graph *cg`)
- void `record_good_nogood` (int good, long double value, `decomposed_tree *child`)
- void `set_good_nogood_trees` (`decomposed_tree *treen`)
- void `rs2tree_filter_domains` (`variables *vars`, `light_stack *value_stack`)
- void `reset_relation_queues` (`set **queues`, int count)
- void `set_cluster_constraints_to_projected_reols_on_c_vars` (`decomposed_tree *treen`)
- `decomposed_tree *` `find_top_root_with_scope` (int \*vars, `decomposed_tree *root`)
- void `assign_table_context` (`llist *usedhts`, `decomposed_tree *treen`)
- int `set_combinations3` (`decomposed_tree *treen`, `set **queues`)
- void `set_queues_treenodes` (`decomposed_tree *treen`, `set **queues`)
- void `set_queues_treenodes2` (`decomposed_tree **treens`, int clusters, `llist *tables`, `set **queues`)
- void `set_queues_treenodes2_singleq` (`decomposed_tree **treens`, int clusters, `llist *tables`, `set **queues`)
- int `process_this_cluster` (`decomposed_tree *current`, `set *undo_set`, `light_stack *undo_stack`, `light_stack *value_stack`, `set **queues`, `decomposed_tree *root`, int \*aqueues\_total\_count, int ppr)
- int `process_tree_combinations3` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, int \*repeat, int isroot, `set **queues`, int queues\_count, int onelevel)
- int `process_tree_combinations5` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations52` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `light_stack *value_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations522` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `light_stack *value_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations53` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations54` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations6` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations7` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations8` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `light_stack *value_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations9` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations4` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `light_stack *value_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations4_` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, `decomposed_tree **treens`, `variable *var`, int ppr)
- int `process_tree_combinations4_filter_up_rec` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, int ppr)
- int `process_tree_combinations4_filter_down_rec` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, int ppr)
- int `process_tree_combinations4_filter` (`decomposed_tree *treen`, `set *undo_set`, `light_stack *undo_stack`, `set **queues`, int queues\_count, int ppr)
- `set *` `collect_htables` (`decomposed_tree *treen`, `set *usedhts`, int separator\_rels\_also)
- `set *` `set_variable_htables` (`decomposed_tree *treen`, `set *usedhts`, int separator\_rels\_also)
- `set *` `set_variable_htables2` (`decomposed_tree **c_array`, int clusters)
- void `generate_combinations_per_cluster` (`decomposed_tree *treen`)
- void `set_default_context_to_tables` (`decomposed_tree *treen`)

- void `report_tree` (`decomposed_tree` \*tree, `rb_tree` \*\*stats, int initial)
- void `add_times_to_stats` (`rb_tree` \*stats)
- void `print_rs2tree_stats` (`rb_tree` \*stats, int count, int thiscluster)
- int `start_w_b_BTD` (`decomposed_tree` \*tree, long double \*total\_sols\_ret, `set` \*queue, `set` \*\*queues, int queues\_count, `decomposed_tree` \*\*treens, int all\_solutions)
- double \* `avgStdev` (`decomposed_tree` \*tree, double \*\*array)
- double \* `avgStdev2` (`rs2tree_stats` \*st, double \*\*array)
- void \* `get_payload` (int param)
- void `traverse_treemap_set_collect_counts` (`tree_map` \*tm, int width, `llist` \*result, double norm\_factor)
- `rs2tree_stats` \* `get_make_stats` (`rb_tree` \*\*stats, `decomposed_tree` \*tree)
- void `init_processed_info` (`rb_tree` \*\*stats, `decomposed_tree` \*tree)
- void `update_processed_info` (`rb_tree` \*\*stats, `decomposed_tree` \*tree)
- void `init_update_processed_info` (`rb_tree` \*\*stats, `decomposed_tree` \*tree, int init, int update)
- void `rs2t_set_heap_pos` (int val, void \*body)
- void `rs2t_set_heap_value` (int val, void \*body)
- int `rs2t_get_heap_value` (void \*body)
- `decomposed_tree` \* `fringe_extract_top` (`heap` \*h)
- int `rs2t_get_id` (void \*body)
- void `check_this` (`set` \*\*queues, int count, int sum)
- void `extract_feature` (`rs2tree_stats` \*rs\_stat, int ppr)
- int `rs2tree_filter_relations_given_domains` (`llist` \*separatorvars, `light_stack` \*undo\_stack, `light_stack` \*value\_stack, `set` \*undo\_set, `set` \*queue, `set` \*\*queues, int \*queues\_total\_count, `llist` \*propagator\_vars)
- void `rs2tree_filter_domains_given_relations` (`llist` \*constraints, `llist` \*separatorvars, `light_stack` \*value\_stack)
- void `rs2tree_filter_domains_given_relation` (`htable` \*rel, `light_stack` \*value\_stack, `llist` \*propagator\_vars)
- int `process_generate_bolstering_bin_clique_relations` (`decomposed_tree` \*root)
- void `pre_feature_extract` (`rb_tree` \*\*stats, `decomposed_tree` \*tree)
- void `post_feature_extract` (`rb_tree` \*stats, `decomposed_tree` \*tree, int number\_of\_clusters)

### 4.59.1 Detailed Description

Implements the interfaces for the structure-based PerTuple and AllSol algorithms.

### 4.59.2 Typedef Documentation

4.59.2.1 typedef struct `rs2tree_configuration` `rs2tree_configuration`

4.59.2.2 typedef struct `rs2tree_stats` `rs2tree_stats`

### 4.59.3 Function Documentation

4.59.3.1 void `add_times_to_stats` ( `rb_tree` \* *stats* )

4.59.3.2 void `assign_table_context` ( `llist` \* *usedhts*, `decomposed_tree` \* *tree* )

4.59.3.3 int `assign_variable` ( `variable` \* *var* )

4.59.3.4 double\* `avgStdev` ( `decomposed_tree` \* *tree*, double \*\* *array* )

4.59.3.5 double\* `avgStdev2` ( `rs2tree_stats` \* *st*, double \*\* *array* )

4.59.3.6 void `check_this` ( `set` \*\* *queues*, int *count*, int *sum* )

4.59.3.7 `variable`\* `choose_var_from_tree` ( `decomposed_tree` \* *tree* )

- 4.59.3.8 `set* collect_htables ( decomposed_tree * treen, set * usedhts, int separator_rels_also )`
- 4.59.3.9 `void extract_feature ( rs2tree_stats * rs_stat, int ppr )`
- 4.59.3.10 `int filter_table_with_chosen_value ( htable * ht, light_stack * undo_stack, set * undo_set, variable * var )`
- 4.59.3.11 `decomposed_tree* find_top_root_with_scope ( int * vars, decomposed_tree * root )`
- 4.59.3.12 `decomposed_tree* fringe_extract_top ( heap * h )`
- 4.59.3.13 `void generate_combinations_per_cluster ( decomposed_tree * treen )`
- 4.59.3.14 `rs2tree_stats* get_make_stats ( rb_tree ** stats, decomposed_tree * treen )`
- 4.59.3.15 `void* get_payload ( int param )`
- 4.59.3.16 `void init_processed_info ( rb_tree ** stats, decomposed_tree * treen )`
- 4.59.3.17 `void init_update_processed_info ( rb_tree ** stats, decomposed_tree * treen, int init, int update )`
- 4.59.3.18 `rs2tree_stats* new_rs2tree_stats ( void )`
- 4.59.3.19 `void post_feature_extract ( rb_tree * stats, decomposed_tree * treen, int number_of_clusters )`
- 4.59.3.20 `void pre_feature_extract ( rb_tree ** stats, decomposed_tree * treen )`
- 4.59.3.21 `void print_rs2tree_stats ( rb_tree * stats, int count, int thiscluster )`
- 4.59.3.22 `int process_geneate_bolstering_bin_clique_relations ( decomposed_tree * root )`
- 4.59.3.23 `int process_this_cluster ( decomposed_tree * current, set * undo_set, light_stack * undo_stack, light_stack * value_stack, set ** queues, decomposed_tree * root, int * aqueues_total_count, int ppr )`
- 4.59.3.24 `int process_tree_combinations ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set * queue, int ppr )`
- 4.59.3.25 `int process_tree_combinations2 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, int * repeat, int isroot, set ** queues, int queues_count, int onelevel, int ppr )`
- 4.59.3.26 `int process_tree_combinations3 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, int * repeat, int isroot, set ** queues, int queues_count, int onelevel )`
- 4.59.3.27 `int process_tree_combinations4 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, light_stack * value_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.28 `int process_tree_combinations4_ ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.29 `int process_tree_combinations4_filter ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, int ppr )`
- 4.59.3.30 `int process_tree_combinations4_filter_down_rec ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, int ppr )`

- 4.59.3.31 `int process_tree_combinations4_filter_up_rec ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, int ppr )`
- 4.59.3.32 `int process_tree_combinations5 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.33 `int process_tree_combinations52 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, light_stack * value_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.34 `int process_tree_combinations522 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, light_stack * value_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.35 `int process_tree_combinations53 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.36 `int process_tree_combinations54 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.37 `int process_tree_combinations6 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.38 `int process_tree_combinations7 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.39 `int process_tree_combinations8 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, light_stack * value_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.40 `int process_tree_combinations9 ( decomposed_tree * treen, set * undo_set, light_stack * undo_stack, set ** queues, int queues_count, decomposed_tree ** treens, variable * var, int ppr )`
- 4.59.3.41 `void record_good_nogood ( int good, long double value, decomposed_tree * child )`
- 4.59.3.42 `void report_tree ( decomposed_tree * treen, rb_tree ** stats, int initial )`
- `for ( np = treen->children->head; np != NULL; np = np->next ) { decomposed_tree * child = (decomposed_tree*)np->body; report_tree ( child, stats, initial, queues ); }`
- 4.59.3.43 `void reset_relation_queues ( set ** queues, int count )`
- 4.59.3.44 `void reset_variable_domain ( variable * var )`
- 4.59.3.45 `void reset_variable_domain2 ( variable * var )`
- 4.59.3.46 `void rs2_all_sols_bt ( decomposed_tree * treen )`
- 4.59.3.47 `int rs2_all_sols_bt_pre_processing ( decomposed_tree * treen, rs2tree_configuration * config, int ppr )`
- 4.59.3.48 `int rs2t_get_heap_value ( void * body )`
- 4.59.3.49 `int rs2t_get_id ( void * body )`
- 4.59.3.50 `void rs2t_set_heap_pos ( int val, void * body )`

- 4.59.3.51 void rs2t\_set\_heap\_value ( int *val*, void \* *body* )
- 4.59.3.52 void rs2tree\_filter\_domains ( variables \* *vars*, light\_stack \* *value\_stack* )
- 4.59.3.53 void rs2tree\_filter\_domains\_given\_relation ( htable \* *rel*, light\_stack \* *value\_stack*, llist \* *propagator\_vars* )
- 4.59.3.54 void rs2tree\_filter\_domains\_given\_relations ( llist \* *constraints*, llist \* *separatorvars*, light\_stack \* *value\_stack* )
- 4.59.3.55 int rs2tree\_filter\_relations\_given\_domains ( llist \* *separatorvars*, light\_stack \* *undo\_stack*, light\_stack \* *value\_stack*, set \* *undo\_set*, set \* *queue*, set \*\* *queues*, int \* *queues\_total\_count*, llist \* *propagator\_vars* )
- 4.59.3.56 void set\_cluster\_constraints\_to\_projected\_reols\_on\_c\_vars ( decomposed\_tree \* *treen* )
- 4.59.3.57 void set\_combinations ( decomposed\_tree \* *treen*, set \*\* *queues* )
- 4.59.3.58 int set\_combinations2 ( decomposed\_tree \* *treen*, set \*\* *queues* )
- 4.59.3.59 int set\_combinations3 ( decomposed\_tree \* *treen*, set \*\* *queues* )
- 4.59.3.60 void set\_default\_context\_to\_tables ( decomposed\_tree \* *treen* )
- 4.59.3.61 void set\_good\_nogood\_trees ( decomposed\_tree \* *treen* )
- 4.59.3.62 void set\_queues\_treenodes ( decomposed\_tree \* *treen*, set \*\* *queues* )
- 4.59.3.63 void set\_queues\_treenodes2 ( decomposed\_tree \*\* *treens*, int *clusters*, llist \* *tables*, set \*\* *queues* )
- 4.59.3.64 void set\_queues\_treenodes2\_singleq ( decomposed\_tree \*\* *treens*, int *clusters*, llist \* *tables*, set \*\* *queues* )
- 4.59.3.65 void set\_single\_graph\_combinations\_to\_treenodes ( decomposed\_tree \* *treen*, llist \* *combs*, constraint\_graph \* *cg* )
- 4.59.3.66 set\* set\_variable\_htables ( decomposed\_tree \* *treen*, set \* *usedhts*, int *separator\_rels\_also* )
- 4.59.3.67 set\* set\_variable\_htables2 ( decomposed\_tree \*\* *c\_array*, int *clusters* )
- 4.59.3.68 void single\_graph\_combinations\_with\_tables ( llist \* *tables*, llist \*\* *ret\_combs*, constraint\_graph \*\* *ret\_cg* )
- 4.59.3.69 int start\_w\_b\_BTD ( decomposed\_tree \* *treen*, long double \* *total\_sols\_ret*, set \* *queue*, set \*\* *queues*, int *queues\_count*, decomposed\_tree \*\* *treens*, int *all\_solutions* )
- 4.59.3.70 int startBTD ( decomposed\_tree \* *treen*, long double \* *total\_sols\_ret*, set \* *queue*, set \*\* *queues*, int *queues\_count*, decomposed\_tree \*\* *treens*, int *print\_sol* )
- 4.59.3.71 void traverse\_treemap\_set\_collect\_counts ( tree\_map \* *tm*, int *width*, llist \* *result*, double *norm\_factor* )
- 4.59.3.72 variable\* un\_assign\_variable ( light\_stack \* *assigned\_var\_stack*, light\_stack \* *undo\_set\_stack*, light\_stack \* *undo\_tuple\_stack*, light\_stack \* *undo\_value\_stack* )
- 4.59.3.73 void undo\_tuple\_deletions\_for\_rs2tree ( set \* *undo\_set*, light\_stack \* *undo\_stack* )
- 4.59.3.74 void undo\_value\_deletions ( light\_stack \* *undo\_value\_stack* )
- 4.59.3.75 void update\_processed\_info ( rb\_tree \*\* *stats*, decomposed\_tree \* *treen* )

## 4.60 septables.c File Reference

```
#include "septables.h"
#include "utils.h"
#include "partition.h"
#include "tables.h"
#include "hashtable_search.h"
#include "active_tuple.h"
#include "rb_tree.h"
```

### Functions

- int [add\\_maxclique\\_scopes](#) ([decomposed\\_tree](#) \*root)
- int [normalize\\_generate\\_separator\\_scopes\\_relations](#) ([decomposed\\_tree](#) \*root)
- int [add\\_binary\\_scopes](#) ([decomposed\\_tree](#) \*root)
- int [shape\\_constraint\\_scopes\\_pE](#) ([decomposed\\_tree](#) \*clique)
- int [normalize\\_small\\_big\\_lists](#) ([llist](#) \*bigger\_new\_constraints, [llist](#) \*smaller\_old\_constraints, [decomposed\\_tree](#) \*clique)
- void [shape\\_constraint\\_scopes\\_pE\\_global](#) ([decomposed\\_tree](#) \*clique)

### Variables

- int [varmap\\_s](#) = 0

### 4.60.1 Function Documentation

4.60.1.1 int [add\\_binary\\_scopes](#) ( [decomposed\\_tree](#) \* *root* )

4.60.1.2 int [add\\_maxclique\\_scopes](#) ( [decomposed\\_tree](#) \* *root* )

4.60.1.3 int [normalize\\_generate\\_separator\\_scopes\\_relations](#) ( [decomposed\\_tree](#) \* *root* )

4.60.1.4 int [normalize\\_small\\_big\\_lists](#) ( [llist](#) \* *bigger\_new\_constraints*, [llist](#) \* *smaller\_old\_constraints*, [decomposed\\_tree](#) \* *clique* )

4.60.1.5 int [shape\\_constraint\\_scopes\\_pE](#) ( [decomposed\\_tree](#) \* *clique* )

4.60.1.6 void [shape\\_constraint\\_scopes\\_pE\\_global](#) ( [decomposed\\_tree](#) \* *clique* )

### 4.60.2 Variable Documentation

4.60.2.1 int [varmap\\_s](#) = 0

## 4.61 septables.h File Reference

```
#include "decomposition.h"
```

### Functions

- int [add\\_maxclique\\_scopes](#) ([decomposed\\_tree](#) \*root)
- int [add\\_binary\\_scopes](#) ([decomposed\\_tree](#) \*root)



- int [shape\\_constraint\\_scopes\\_pE](#) ([decomposed\\_tree](#) \*clique)
- void [shape\\_constraint\\_scopes\\_pE\\_global](#) ([decomposed\\_tree](#) \*clique)
- int [normalize\\_generate\\_separator\\_scopes\\_relations](#) ([decomposed\\_tree](#) \*root)
- int [normalize\\_small\\_big\\_lists](#) ([llist](#) \*bigger\_new\_constraints, [llist](#) \*smaller\_old\_constraints, [decomposed\\_tree](#) \*clique)

### 4.61.1 Detailed Description

Implements helper methods for generating redundant constraints at the separators.

### 4.61.2 Function Documentation

4.61.2.1 int [add\\_binary\\_scopes](#) ( [decomposed\\_tree](#) \* root )

4.61.2.2 int [add\\_maxclique\\_scopes](#) ( [decomposed\\_tree](#) \* root )

4.61.2.3 int [normalize\\_generate\\_separator\\_scopes\\_relations](#) ( [decomposed\\_tree](#) \* root )

4.61.2.4 int [normalize\\_small\\_big\\_lists](#) ( [llist](#) \* *bigger\_new\_constraints*, [llist](#) \* *smaller\_old\_constraints*, [decomposed\\_tree](#) \* *clique* )

4.61.2.5 int [shape\\_constraint\\_scopes\\_pE](#) ( [decomposed\\_tree](#) \* *clique* )

4.61.2.6 void [shape\\_constraint\\_scopes\\_pE\\_global](#) ( [decomposed\\_tree](#) \* *clique* )

## 4.62 set.c File Reference

```
#include "set.h"
#include "utils.h"
#include "rb_tree.h"
```

### Functions

- [s\\_node](#) \* [new\\_s\\_node](#) (int key, void \*body)
- [set](#) \* [copy\\_set](#) ([set](#) \*s)
- [set](#) \* [new\\_set](#) (int size)
- [set](#) \* [new\\_set\\_nomap](#) (int size)
- void [reset\\_set](#) ([set](#) \*s)
- [set](#) \* [new\\_set\\_map](#) (int size)
- [llist](#) \* [set\\_to\\_list](#) ([set](#) \*s)
- void [destroy\\_set](#) ([set](#) \*s)
- int [add\\_int\\_element](#) ([set](#) \*s, int key)
- int [get\\_int\\_element](#) ([set](#) \*s, int key)
- int [remove\\_int\\_element](#) ([set](#) \*s, int key)
- int [add\\_element](#) ([set](#) \*s, void \*e, int key)
- int [add\\_element\\_nomap](#) ([set](#) \*s, void \*e, int key)
- void [add\\_element\\_node](#) ([set](#) \*s, [s\\_node](#) \*n)
- void \* [remove\\_element](#) ([set](#) \*s)
- void \* [peek\\_element](#) ([set](#) \*s)
- void \* [remove\\_element\\_with\\_key](#) ([set](#) \*s, int key, int [get\\_key](#)(void \*))
- void \* [get\\_element\\_with\\_key](#) ([set](#) \*s, int key, int [get\\_key](#)(void \*))
- int [remove\\_element\\_key](#) ([set](#) \*s)
- int [set\\_empty](#) ([set](#) \*s)

### 4.62.1 Function Documentation

4.62.1.1 `int add_element ( set * s, void * e, int key )`

4.62.1.2 `void add_element_node ( set * s, s_node * n )`

4.62.1.3 `int add_element_nomap ( set * s, void * e, int key )`

4.62.1.4 `int add_int_element ( set * s, int key )`

4.62.1.5 `set* copy_set ( set * s )`

4.62.1.6 `void destroy_set ( set * s )`

4.62.1.7 `void* get_element_with_key ( set * s, int key, int get_keyvoid * )`

4.62.1.8 `int get_int_element ( set * s, int key )`

4.62.1.9 `s_node* new_s_node ( int key, void * body )`

4.62.1.10 `set* new_set ( int size )`

4.62.1.11 `set* new_set_map ( int size )`

4.62.1.12 `set* new_set_nomap ( int size )`

4.62.1.13 `void* peek_element ( set * s )`

4.62.1.14 `void* remove_element ( set * s )`

4.62.1.15 `int remove_element_key ( set * s )`

4.62.1.16 `void* remove_element_with_key ( set * s, int key, int get_keyvoid * )`

4.62.1.17 `int remove_int_element ( set * s, int key )`

4.62.1.18 `void reset_set ( set * s )`

4.62.1.19 `int set_empty ( set * s )`

4.62.1.20 `llist* set_to_list ( set * s )`

### 4.63 set.h File Reference

```
#include <stdlib.h>
#include "llist.h"
```

#### Classes

- struct [s\\_node](#)
- struct [set](#)

## Typedefs

- typedef struct [s\\_node](#) [s\\_node](#)
- typedef struct [set](#) [set](#)

## Functions

- [s\\_node](#) \* [new\\_s\\_node](#) (int key, void \*body)
- [set](#) \* [new\\_set](#) (int size)
- [set](#) \* [new\\_set\\_map](#) (int size)
- [set](#) \* [new\\_set\\_nomap](#) (int size)
- int [add\\_element](#) ([set](#) \*s, void \*e, int key)
- int [add\\_element\\_nomap](#) ([set](#) \*s, void \*e, int key)
- void [add\\_element\\_node](#) ([set](#) \*s, [s\\_node](#) \*n)
- void \* [remove\\_element](#) ([set](#) \*s)
- int [remove\\_element\\_key](#) ([set](#) \*s)
- int [set\\_empty](#) ([set](#) \*s)
- void [destroy\\_set](#) ([set](#) \*s)
- void \* [peek\\_element](#) ([set](#) \*s)
- void \* [remove\\_element\\_with\\_key](#) ([set](#) \*s, int key, int [get\\_key](#)(void \*))
- void \* [get\\_element\\_with\\_key](#) ([set](#) \*s, int key, int [get\\_key](#)(void \*))
- int [remove\\_int\\_element](#) ([set](#) \*s, int key)
- int [get\\_int\\_element](#) ([set](#) \*s, int key)
- int [add\\_int\\_element](#) ([set](#) \*s, int key)
- [set](#) \* [copy\\_set](#) ([set](#) \*s)
- void [reset\\_set](#) ([set](#) \*s)
- [llist](#) \* [set\\_to\\_list](#) ([set](#) \*s)

### 4.63.1 Detailed Description

Provides a naive set implementation.

### 4.63.2 Typedef Documentation

4.63.2.1 typedef struct [s\\_node](#) [s\\_node](#)

4.63.2.2 typedef struct [set](#) [set](#)

### 4.63.3 Function Documentation

4.63.3.1 int [add\\_element](#) ( [set](#) \* s, void \* e, int key )

4.63.3.2 void [add\\_element\\_node](#) ( [set](#) \* s, [s\\_node](#) \* n )

4.63.3.3 int [add\\_element\\_nomap](#) ( [set](#) \* s, void \* e, int key )

4.63.3.4 int [add\\_int\\_element](#) ( [set](#) \* s, int key )

4.63.3.5 [set](#)\* [copy\\_set](#) ( [set](#) \* s )

4.63.3.6 void [destroy\\_set](#) ( [set](#) \* s )

4.63.3.7 void\* [get\\_element\\_with\\_key](#) ( [set](#) \* s, int key, int [get\\_key](#)void \* )

- 4.63.3.8 `int get_int_element ( set * s, int key )`
- 4.63.3.9 `s_node* new_s_node ( int key, void * body )`
- 4.63.3.10 `set* new_set ( int size )`
- 4.63.3.11 `set* new_set_map ( int size )`
- 4.63.3.12 `set* new_set_nomap ( int size )`
- 4.63.3.13 `void* peek_element ( set * s )`
- 4.63.3.14 `void* remove_element ( set * s )`
- 4.63.3.15 `int remove_element_key ( set * s )`
- 4.63.3.16 `void* remove_element_with_key ( set * s, int key, int get_keyvoid * )`
- 4.63.3.17 `int remove_int_element ( set * s, int key )`
- 4.63.3.18 `void reset_set ( set * s )`
- 4.63.3.19 `int set_empty ( set * s )`
- 4.63.3.20 `llist* set_to_list ( set * s )`

## 4.64 solution.c File Reference

```
#include "solution.h"
#include <stdio.h>
#include "utils.h"
#include "llist.h"
```

### Functions

- `solution * read_solution (char *filename)`
- `int check_in_solution (htable *ht, int t, solution *sol)`
- `int table_has_the_solution (htable *ht, solution *sol)`
- `int assignment_has_the_solution (variables *vars, solution *sol)`
- `void confirm_variables (variables *vars, solution *sol)`

#### 4.64.1 Function Documentation

- 4.64.1.1 `int assignment_has_the_solution ( variables * vars, solution * sol )`
- 4.64.1.2 `int check_in_solution ( htable * ht, int t, solution * sol )`
- 4.64.1.3 `void confirm_variables ( variables * vars, solution * sol )`
- 4.64.1.4 `solution* read_solution ( char * filename )`
- 4.64.1.5 `int table_has_the_solution ( htable * ht, solution * sol )`

## 4.65 solution.h File Reference

```
#include "hashtable.h"
#include "variables.h"
```

### Classes

- struct [solution](#)

### Typedefs

- typedef struct [solution](#) [solution](#)

### Functions

- [solution](#) \* [read\\_solution](#) (char \*filename)
- int [check\\_in\\_solution](#) (htable \*ht, int t, [solution](#) \*sol)
- int [table\\_has\\_the\\_solution](#) (htable \*ht, [solution](#) \*sol)
- int [assignment\\_has\\_the\\_solution](#) (variables \*vars, [solution](#) \*sol)
- void [confirm\\_variables](#) (variables \*vars, [solution](#) \*sol)

#### 4.65.1 Detailed Description

Represents a user-given solution and method for querying it. Mainly used for debugging.

#### 4.65.2 Typedef Documentation

##### 4.65.2.1 typedef struct [solution](#) [solution](#)

#### 4.65.3 Function Documentation

##### 4.65.3.1 int [assignment\\_has\\_the\\_solution](#) ( variables \* vars, solution \* sol )

##### 4.65.3.2 int [check\\_in\\_solution](#) ( htable \* ht, int t, solution \* sol )

##### 4.65.3.3 void [confirm\\_variables](#) ( variables \* vars, solution \* sol )

##### 4.65.3.4 [solution](#)\* [read\\_solution](#) ( char \* filename )

##### 4.65.3.5 int [table\\_has\\_the\\_solution](#) ( htable \* ht, solution \* sol )

## 4.66 sort.c File Reference

```
#include "utils.h"
#include "sort.h"
#include <stdlib.h>
#include <stdio.h>
#include "hashtable.h"
```

## Functions

- void `buble_sort_list` (`llist *tosort`, int `get_value`(void \*, void \*), void \*helper)
- void `buble_sort_list_2` (`llist *tosort`, int `compare`(void \*, void \*))
- void `buble_sort` (int \*a)
- void `swap_int` (int i, int j, void \*a, int width, int `get_value`(void \*, int, int), void `set_value`(void \*, int, int, int))
- void `swap_obj` (int i, int j, void \*a, int width, void \*`get_object`(void \*, int, int), void `set_object`(void \*, int, int, void \*))
- int `is_sorted` (void \*a, int start, int end, int key\_col, int `get_value`(void \*, int, int))
- void `sort_relative_to_col` (void \*a, int size, int width, int key\_col, int key\_col2, `llist_node *cols`, int `get_value`(void \*, int, int), void \*`get_object`(void \*, int, int), void `set_value`(void \*, int, int, int), void `set_object`(void \*, int, int, void \*), int setobject)
- void `sort_w_obj_mcols` (void \*a, int size, int width, `llist *cols`, int `get_value`(void \*, int, int), void \*`get_object`(void \*, int, int), void `set_value`(void \*, int, int, int), void `set_object`(void \*, int, int, void \*), int setobject)
- void `sort_w_obj` (void \*a, int size, int width, int key\_col, int key\_col2, int `get_value`(void \*, int, int), void \*`get_object`(void \*, int, int), void `set_value`(void \*, int, int, int), void `set_object`(void \*, int, int, void \*), int setobject)
- void `sort_w_mcols` (void \*a, int size, int width, `llist *cols`, int `get_value`(void \*, int, int), void `set_value`(void \*, int, int, int))
- void `sort_w` (void \*a, int size, int width, int key\_col, int key\_col2, int `get_value`(void \*, int, int), void `set_value`(void \*, int, int, int))
- int `get_rand_value` (void \*av, int i, int c)
- void `set_rand_value` (void \*av, int i, int c, int v)
- void `sort_recursive` (void \*a, int p, int r, int key\_col, int width, int randp, int `get_value`(void \*, int, int), void \*`get_object`(void \*, int, int), void `set_value`(void \*, int, int, int), void `set_object`(void \*, int, int, void \*), int setobject)

### 4.66.1 Function Documentation

4.66.1.1 void `buble_sort` ( int \* a )

4.66.1.2 void `buble_sort_list` ( `llist * tosort`, int `get_value`void \*, void \*, void \* helper )

4.66.1.3 void `buble_sort_list_2` ( `llist * tosort`, int `compare`void \*, void \* )

4.66.1.4 int `get_rand_value` ( void \* av, int i, int c )

4.66.1.5 int `is_sorted` ( void \* a, int start, int end, int key\_col, int `get_value`void \*, int, int )

4.66.1.6 void `set_rand_value` ( void \* av, int i, int c, int v )

4.66.1.7 void `sort_recursive` ( void \* a, int p, int r, int key\_col, int width, int randp, int `get_value`void \*, int, int, void \* `get_object`void \*, int, int, void `set_value`void \*, int, int, int, void `set_object`void \*, int, int, void \*, int setobject )

4.66.1.8 void `sort_relative_to_col` ( void \* a, int size, int width, int key\_col, int key\_col2, `llist_node * cols`, int `get_value`void \*, int, int, void \* `get_object`void \*, int, int, void `set_value`void \*, int, int, int, void `set_object`void \*, int, int, void \*, int setobject )

4.66.1.9 void `sort_w` ( void \* a, int size, int width, int key\_col, int key\_col2, int `get_value`void \*, int, int, void `set_value`void \*, int, int, int )

4.66.1.10 void `sort_w_mcols` ( void \* a, int size, int width, `llist * cols`, int `get_value`void \*, int, int, void `set_value`void \*, int, int, int )

4.66.1.11 void `sort_w_obj` ( void \* a, int size, int width, int key\_col, int key\_col2, int `get_value`void \*, int, int, void \* `get_object`void \*, int, int, void `set_value`void \*, int, int, int, void `set_object`void \*, int, int, void \*, int setobject )

4.66.1.12 void `sort_w_obj_mcols` ( void \* *a*, int *size*, int *width*, llist \* *cols*, int *get\_valuevoid* \*, int, int, void \* *get\_objectvoid* \*, int, int, void *set\_valuevoid* \*, int, int, int, void *set\_objectvoid* \*, int, int, void \*, int *setobject* )

4.66.1.13 void `swap_int` ( int *i*, int *j*, void \* *a*, int *width*, int *get\_valuevoid* \*, int, int, void *set\_valuevoid* \*, int, int, int )

4.66.1.14 void `swap_obj` ( int *i*, int *j*, void \* *a*, int *width*, void \* *get\_objectvoid* \*, int, int, void *set\_objectvoid* \*, int, int, void \* )

## 4.67 sort.h File Reference

### Functions

- void `buble_sort_list` (llist \**tosort*, int *get\_value*(void \*, void \*), void \**helper*)
- void `buble_sort_list_2` (llist \**tosort*, int *compare*(void \*, void \*\*))
- void `buble_sort` (int \**a*)
- void `sort` (void \**a*, int *size*, int *key\_col*, int *key\_col2*, int *get\_value*(), void *set\_object*(void \*, int, int, void \*\*))
- void `sort_w` (void \**a*, int *size*, int *width*, int *key\_col*, int *key\_col2*, int *get\_value*(void \*, int, int), void *set\_value*(void \*, int, int, int))
- void `sort_relative_to_col` (void \**a*, int *size*, int *width*, int *key\_col*, int *key\_col2*, llist\_node \**cols*, int *get\_value*(void \*, int, int), void \**get\_object*(void \*, int, int), void *set\_value*(void \*, int, int, int), void *set\_object*(void \*, int, int, void \*\*), int *setobject*)
- void `sort_w_obj` (void \**a*, int *size*, int *width*, int *key\_col*, int *key\_col2*, int *get\_value*(void \*, int, int), void \**get\_object*(void \*, int, int), void *set\_value*(void \*, int, int, int), void *set\_object*(void \*, int, int, void \*\*), int *setobject*)
- int `get_rand_value` (void \**av*, int *i*, int *c*)
- void `set_rand_value` (void \**av*, int *i*, int *c*, int *v*)
- void `sort_recursive` (void \**a*, int *p*, int *r*, int *key\_col*, int *width*, int *randp*, int *get\_value*(void \*, int, int), void \**get\_object*(void \*, int, int), void *set\_value*(void \*, int, int, int), void *set\_object*(void \*, int, int, void \*\*), int *setobject*)
- void `sort_w_obj_mcols` (void \**a*, int *size*, int *width*, llist \**cols*, int *get\_value*(void \*, int, int), void \**get\_object*(void \*, int, int), void *set\_value*(void \*, int, int, int), void *set\_object*(void \*, int, int, void \*\*), int *setobject*)
- void `sort_w_mcols` (void \**a*, int *size*, int *width*, llist \**cols*, int *get\_value*(void \*, int, int), void *set\_value*(void \*, int, int, int))
- void `swap_int` (int *i*, int *j*, void \**a*, int *width*, int *get\_value*(void \*, int, int), void *set\_value*(void \*, int, int, int))
- void `swap_obj` (int *i*, int *j*, void \**a*, int *width*, void \**get\_object*(void \*, int, int), void *set\_object*(void \*, int, int, void \*\*))
- int `is_sorted` (void \**a*, int *start*, int *end*, int *key\_col*, int *get\_value*())

### 4.67.1 Detailed Description

Provides an implementation for quick-sort.

### 4.67.2 Function Documentation

4.67.2.1 void `buble_sort` ( int \* *a* )

4.67.2.2 void `buble_sort_list` ( llist \* *tosort*, int *get\_valuevoid* \*, void \*, void \* *helper* )

4.67.2.3 void `buble_sort_list_2` ( llist \* *tosort*, int *comparevoid* \*, void \* )

4.67.2.4 int `get_rand_value` ( void \* *av*, int *i*, int *c* )

4.67.2.5 int `is_sorted` ( void \* *a*, int *start*, int *end*, int *key\_col*, int *get\_value*() )

4.67.2.6 void `set_rand_value` ( void \* *av*, int *i*, int *c*, int *v* )

- 4.67.2.7 void sort ( void \* a, int size, int key\_col, int key\_col2, int get\_value(), void set\_objectvoid \*, int, int, void \* )
- 4.67.2.8 void sort\_recursive ( void \* a, int p, int r, int key\_col, int width, int randp, int get\_valuevoid \*, int, int, void \* get\_objectvoid \*, int, int, void set\_valuevoid \*, int, int, int, void set\_objectvoid \*, int, int, void \*, int setobject )
- 4.67.2.9 void sort\_relative\_to\_col ( void \* a, int size, int width, int key\_col, int key\_col2, llist\_node \* cols, int get\_valuevoid \*, int, int, void \* get\_objectvoid \*, int, int, void set\_valuevoid \*, int, int, int, void set\_objectvoid \*, int, int, void \*, int setobject )
- 4.67.2.10 void sort\_w ( void \* a, int size, int width, int key\_col, int key\_col2, int get\_valuevoid \*, int, int, void set\_valuevoid \*, int, int, int )
- 4.67.2.11 void sort\_w\_mcols ( void \* a, int size, int width, llist \* cols, int get\_valuevoid \*, int, int, void set\_valuevoid \*, int, int, int )
- 4.67.2.12 void sort\_w\_obj ( void \* a, int size, int width, int key\_col, int key\_col2, int get\_valuevoid \*, int, int, void \* get\_objectvoid \*, int, int, void set\_valuevoid \*, int, int, int, void set\_objectvoid \*, int, int, void \*, int setobject )
- 4.67.2.13 void sort\_w\_obj\_mcols ( void \* a, int size, int width, llist \* cols, int get\_valuevoid \*, int, int, void \* get\_objectvoid \*, int, int, void set\_valuevoid \*, int, int, int, void set\_objectvoid \*, int, int, void \*, int setobject )
- 4.67.2.14 void swap\_int ( int i, int j, void \* a, int width, int get\_valuevoid \*, int, int, void set\_valuevoid \*, int, int, int )
- 4.67.2.15 void swap\_obj ( int i, int j, void \* a, int width, void \* get\_objectvoid \*, int, int, void set\_objectvoid \*, int, int, void \* )

## 4.68 tree\_map.c File Reference

```
#include "tree_map.h"
```

### Functions

- [tree\\_map \\* new\\_tree\\_map](#) (htable \*http, llist \*common\_cols, int skip\_deleted)
- [void compress\\_tree\\_map](#) (tree\_map \*tm)
- [tree\\_map\\_node \\* compress\\_tree\\_map\\_rec](#) (llist \*ll, llist\_node \*level)
- [int \\*\\* compress\\_tuple\\_list](#) (llist \*ll)
- [tree\\_map\\_node \\* compress\\_tree\\_map\\_node\\_list](#) (llist \*ll)
- [tree\\_map\\_node \\* find\\_node\\_with\\_id\\_compressesd](#) (int id, tree\_map\_node \*al, int size, int search, int wildcard)
- [tree\\_map\\_node \\* find\\_node\\_with\\_id](#) (int id, llist \*l, int search, int wildcard)
- [tree\\_map\\_node \\* new\\_tree\\_map\\_node](#) (int id)
- [int get\\_tuples\\_for\\_tuple\\_compressed](#) (int \*tuple, tree\_map \*tm, llist \*common\_cols, int wildcard, int \*\*\*result)
- [llist \\* get\\_tuples\\_for\\_tuple](#) (int \*tuple, tree\_map \*tm, llist \*common\_cols, int wildcard)
- [int insert\\_tuple\\_into\\_tree](#) (int \*tuple, tree\_map \*tm, llist \*common\_cols, int wildcard)
- [int insert\\_tuple\\_into\\_tree\\_payload](#) (int \*tuple, tree\_map \*tm, llist \*common\_cols, int wildcard, void \*payload)
- [htable \\* tree\\_map\\_to\\_htable\\_rec](#) (tree\_map \*tm, char \*name, int id, char \*\*var\_names, int \*vars)
- [void fetch\\_tuples](#) (tree\_map\_node \*tmn, htable \*ht, int \*tuple, int depth, int length)
- [void reset\\_tree\\_map](#) (tree\_map \*tm)
- [void destroy\\_tree\\_map](#) (tree\_map \*tm, int payload)
- [void destroy\\_tree\\_map\\_lists](#) (llist \*l, int level, int limit, int destroy\_payload)
- [void destroy\\_tree\\_map\\_arrays](#) (tree\_map\_node \*la, int level, int limit, int destroy\_payload, int size)
- [htable \\* tree\\_map\\_to\\_htable](#) (tree\_map \*tm, char \*name, int id, char \*\*var\_names, int \*vars, htable \*tofill)
- [void traverse\\_treemap](#) (tree\_map \*tm)



### 4.68.1 Function Documentation

- 4.68.1.1 void compress\_tree\_map ( tree\_map \* tm )
- 4.68.1.2 tree\_map\_node\* compress\_tree\_map\_node\_list ( llist \* ll )
- 4.68.1.3 tree\_map\_node\* compress\_tree\_map\_rec ( llist \* ll, llist\_node \* level )
- 4.68.1.4 int\*\* compress\_tuple\_list ( llist \* ll )
- 4.68.1.5 void destroy\_tree\_map ( tree\_map \* tm, int payload )
- 4.68.1.6 void destroy\_tree\_map\_arrays ( tree\_map\_node \* la, int level, int limit, int destroy\_payload, int size )
- 4.68.1.7 void destroy\_tree\_map\_lists ( llist \* l, int level, int limit, int destroy\_payload )
- 4.68.1.8 void fetch\_tuples ( tree\_map\_node \* tmn, htable \* ht, int \* tuple, int depth, int length )
- 4.68.1.9 tree\_map\_node\* find\_node\_with\_id ( int id, llist \* l, int search, int wildcard )
- 4.68.1.10 tree\_map\_node\* find\_node\_with\_id\_compressed ( int id, tree\_map\_node \* al, int size, int search, int wildcard )
- 4.68.1.11 llist\* get\_tuples\_for\_tuple ( int \* tuple, tree\_map \* tm, llist \* common\_cols, int wildcard )
- 4.68.1.12 int get\_tuples\_for\_tuple\_compressed ( int \* tuple, tree\_map \* tm, llist \* common\_cols, int wildcard, int \*\*\* result )
- 4.68.1.13 int insert\_tuple\_into\_tree ( int \* tuple, tree\_map \* tm, llist \* common\_cols, int wildcard )
- 4.68.1.14 int insert\_tuple\_into\_tree\_payload ( int \* tuple, tree\_map \* tm, llist \* common\_cols, int wildcard, void \* payload )
- 4.68.1.15 tree\_map\* new\_tree\_map ( htable \* htto, llist \* common\_cols, int skip\_deleted )
- 4.68.1.16 tree\_map\_node\* new\_tree\_map\_node ( int id )
- 4.68.1.17 void reset\_tree\_map ( tree\_map \* tm )
- 4.68.1.18 void traverse\_treemap ( tree\_map \* tm )
- 4.68.1.19 htable\* tree\_map\_to\_htable ( tree\_map \* tm, char \* name, int id, char \*\* var\_names, int \* vars, htable \* tofill )
- 4.68.1.20 htable\* tree\_map\_to\_htable\_rec ( tree\_map \* tm, char \* name, int id, char \*\* var\_names, int \* vars )

## 4.69 tree\_map.h File Reference

```
#include "utils.h"
#include "hashtable.h"
#include "llist.h"
```

### Classes

- struct [tree\\_map](#)
- struct [tree\\_map\\_node](#)

## Macros

- `#define STAR -99`

## Typedefs

- `typedef struct tree_map tree_map`
- `typedef struct tree_map_node tree_map_node`

## Functions

- `tree_map * new_tree_map (htable *h, llist *common_cols, int skip_deleted)`
- `tree_map_node * find_node_with_id (int id, llist *l, int search, int wildcard)`
- `tree_map_node * find_node_with_id_compressed (int id, tree_map_node *al, int size, int search, int wildcard)`
- `tree_map_node * new_tree_map_node (int id)`
- `llist * get_tuples_for_tuple (int *tuple, tree_map *tm, llist *common_cols, int wildcard)`
- `int get_tuples_for_tuple_compressed (int *tuple, tree_map *tm, llist *common_cols, int wildcard, int **result)`
- `int insert_tuple_into_tree_payload (int *tuple, tree_map *tm, llist *common_cols, int wildcard, void *payload)`
- `int insert_tuple_into_tree (int *tuple, tree_map *tm, llist *common_cols, int wildcard)`
- `htable * tree_map_to_htable (tree_map *tm, char *name, int id, char **var_names, int *vars, htable *tofill)`
- `htable * tree_map_to_htable_rec (tree_map *tm, char *name, int id, char **var_names, int *vars)`
- `void fetch_tuples (tree_map_node *tmn, htable *ht, int *tuple, int depth, int length)`
- `void destroy_tree_map (tree_map *tm, int payload)`
- `void destroy_tree_map_lists (llist *l, int level, int limit, int destroy_payload)`
- `void destroy_tree_map_arrays (tree_map_node *la, int level, int limit, int destroy_payload, int size)`
- `void traverse_treemap (tree_map *tm)`
- `void reset_tree_map (tree_map *tm)`
- `tree_map_node * compress_tree_map_rec (llist *ll, llist_node *level)`
- `void compress_tree_map (tree_map *tm)`
- `int ** compress_tuple_list (llist *ll)`
- `tree_map_node * compress_tree_map_node_list (llist *ll)`

### 4.69.1 Detailed Description

Implements the index-tree data structure for finding matching tuples in relations. Also provides compression algorithms.

### 4.69.2 Macro Definition Documentation

#### 4.69.2.1 `#define STAR -99`

### 4.69.3 Typedef Documentation

#### 4.69.3.1 `typedef struct tree_map tree_map`

#### 4.69.3.2 `typedef struct tree_map_node tree_map_node`

### 4.69.4 Function Documentation

#### 4.69.4.1 `void compress_tree_map ( tree_map * tm )`

- 4.69.4.2 `tree_map_node* compress_tree_map_node_list ( llist * ll )`
- 4.69.4.3 `tree_map_node* compress_tree_map_rec ( llist * ll, llist_node * level )`
- 4.69.4.4 `int** compress_tuple_list ( llist * ll )`
- 4.69.4.5 `void destroy_tree_map ( tree_map * tm, int payload )`
- 4.69.4.6 `void destroy_tree_map_arrays ( tree_map_node * la, int level, int limit, int destroy_payload, int size )`
- 4.69.4.7 `void destroy_tree_map_lists ( llist * l, int level, int limit, int destroy_payload )`
- 4.69.4.8 `void fetch_tuples ( tree_map_node * tmn, htable * ht, int * tuple, int depth, int length )`
- 4.69.4.9 `tree_map_node* find_node_with_id ( int id, llist * l, int search, int wildcard )`
- 4.69.4.10 `tree_map_node* find_node_with_id_compressed ( int id, tree_map_node * al, int size, int search, int wildcard )`
- 4.69.4.11 `llist* get_tuples_for_tuple ( int * tuple, tree_map * tm, llist * common_cols, int wildcard )`
- 4.69.4.12 `int get_tuples_for_tuple_compressed ( int * tuple, tree_map * tm, llist * common_cols, int wildcard, int *** result )`
- 4.69.4.13 `int insert_tuple_into_tree ( int * tuple, tree_map * tm, llist * common_cols, int wildcard )`
- 4.69.4.14 `int insert_tuple_into_tree_payload ( int * tuple, tree_map * tm, llist * common_cols, int wildcard, void * payload )`
- 4.69.4.15 `tree_map* new_tree_map ( htable * htto, llist * common_cols, int skip_deleted )`
- 4.69.4.16 `tree_map_node* new_tree_map_node ( int id )`
- 4.69.4.17 `void reset_tree_map ( tree_map * tm )`
- 4.69.4.18 `void traverse_treemap ( tree_map * tm )`
- 4.69.4.19 `htable* tree_map_to_htable ( tree_map * tm, char * name, int id, char ** var_names, int * vars, htable * tofill )`
- 4.69.4.20 `htable* tree_map_to_htable_rec ( tree_map * tm, char * name, int id, char ** var_names, int * vars )`

## 4.70 utils.c File Reference

```
#include <execinfo.h>
#include <malloc.h>
#include "utils.h"
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "llist.h"
#include "globals.h"
```

### Macros

- #define `PROFILE_SIZE` 2000

## Functions

- void [profile\\_time\\_start](#) (int id, const char \*desc)
- double \* [get\\_profile\\_times](#) (void)
- void [profile\\_time\\_end](#) (int id)
- void [profile\\_crash\\_all\\_end](#) (void)
- void [print\\_profile](#) (void)
- int [string\\_equal\\_casted](#) (void \*a, void \*b)
- int [string\\_equal](#) (char \*a, char \*b)
- char \* [str\\_copy](#) (char \*a)
- int [compare\\_ints](#) (const void \*a, const void \*b)
- char \* [str\\_copy\\_short](#) (char \*a, int max)
- int [max](#) (int a, int b)
- char \* [str\\_cat](#) (char \*a, const char \*b)
- char \* [str\\_cat\\_sep](#) (char \*a, char \*b, char \*s)
- void [warn](#) (const char \*a)
- void [error](#) (const char \*a)
- char \* [str\\_cat\\_recursive](#) (char \*a, char \*b, const char \*s)
- void \* [checked\\_malloc](#) (int size)
- void [checked\\_free](#) (void \*buffer)
- int [skip\\_char](#) (char \*line, int i, char c)
- int [skip\\_space](#) (char \*line, int i)
- int [get\\_int](#) (char \*line, int i, int \*num)
- int [get\\_double](#) (char \*line, int i, double \*num)
- int [skip\\_to\\_char](#) (char \*line, int i, char c)
- int [get\\_string](#) (char \*line, int i, char \*\*str)
- int [visit\\_var](#) (int var)
- void [unvisit\\_var](#) (int var)
- int [set\\_var](#) (int var)
- void [reset\\_vars](#) (void)
- void [print\\_trace](#) (void)
- void [catcher](#) (int sig)
- void [dump\\_trace](#) (void)

## Variables

- double [profiled](#) [[PROFILE\\_SIZE](#)]
- char \* [profile\\_desc](#) [[PROFILE\\_SIZE](#)]
- int [profiled\\_count](#) [[PROFILE\\_SIZE](#)]

### 4.70.1 Macro Definition Documentation

4.70.1.1 `#define PROFILE_SIZE 2000`

### 4.70.2 Function Documentation

4.70.2.1 `void catcher ( int sig )`

4.70.2.2 `void checked_free ( void * buffer )`

4.70.2.3 `void* checked_malloc ( int size )`

4.70.2.4 `int compare_ints ( const void * a, const void * b )`

- 4.70.2.5 void dump\_trace ( void )
- 4.70.2.6 void error ( const char \* a )
- 4.70.2.7 int get\_double ( char \* line, int i, double \* num )
- 4.70.2.8 int get\_int ( char \* line, int i, int \* num )
- 4.70.2.9 double\* get\_profile\_times ( void )
- 4.70.2.10 int get\_string ( char \* line, int i, char \*\* str )
- 4.70.2.11 int max ( int a, int b )
- 4.70.2.12 void print\_profile ( void )
- 4.70.2.13 void print\_trace ( void )
- 4.70.2.14 void profile\_crash\_all\_end ( void )
- 4.70.2.15 void profile\_time\_end ( int id )
- 4.70.2.16 void profile\_time\_start ( int id, const char \* desc )
- 4.70.2.17 void reset\_vars ( void )
- 4.70.2.18 int set\_var ( int var )
- 4.70.2.19 int skip\_char ( char \* line, int i, char c )
- 4.70.2.20 int skip\_space ( char \* line, int i )
- 4.70.2.21 int skip\_to\_char ( char \* line, int i, char c )
- 4.70.2.22 char\* str\_cat ( char \* a, const char \* b )
- 4.70.2.23 char\* str\_cat\_recursive ( char \* a, char \* b, const char \* s )
- 4.70.2.24 char\* str\_cat\_sep ( char \* a, char \* b, char \* s )
- 4.70.2.25 char\* str\_copy ( char \* a )
- 4.70.2.26 char\* str\_copy\_short ( char \* a, int max )
- 4.70.2.27 int string\_equal ( char \* a, char \* b )
- 4.70.2.28 int string\_equal\_casted ( void \* a, void \* b )
- 4.70.2.29 void unvisit\_var ( int var )
- 4.70.2.30 int visit\_var ( int var )
- 4.70.2.31 void warn ( const char \* a )

### 4.70.3 Variable Documentation

4.70.3.1 `char* profile_desc[PROFILE_SIZE]`

4.70.3.2 `double profiled[PROFILE_SIZE]`

4.70.3.3 `int profiled_count[PROFILE_SIZE]`

## 4.71 `utils.h` File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include "globals.h"
```

### Functions

- void `profile_time_start` (int id, const char \*desc)
- double \* `get_profile_times` (void)
- void `profile_time_end` (int id)
- void `print_profile` (void)
- int `string_equal_casted` (void \*a, void \*b)
- int `string_equal` (char \*a, char \*b)
- char \* `str_copy` (char \*a)
- char \* `str_copy_short` (char \*a, int max)
- char \* `str_cat` (char \*a, const char \*b)
- char \* `str_cat_sep` (char \*a, char \*b, char \*s)
- char \* `str_cat_recursive` (char \*a, char \*b, const char \*s)
- void `warn` (const char \*a)
- void `error` (const char \*a)
- void \* `checked_malloc` (int size)
- void `checked_free` (void \*buffer)
- int `max` (int a, int b)
- int `compare_ints` (const void \*a, const void \*b)
- int `skip_char` (char \*line, int i, char c)
- int `skip_space` (char \*line, int i)
- int `get_int` (char \*line, int i, int \*num)
- int `get_double` (char \*line, int i, double \*num)
- int `skip_to_char` (char \*line, int i, char c)
- int `get_string` (char \*line, int i, char \*\*str)
- int `set_var` (int var)
- void `reset_vars` (void)
- int `visit_var` (int var)
- void `unvisit_var` (int var)
- void `print_trace` (void)
- void `print_allocs` (void)
- void `profile_crash_all_end` (void)
- void `catcher` (int sig)
- void `dump_trace` (void)

### 4.71.1 Detailed Description

Implements various utilities, such as code profiling tools.

## 4.71.2 Function Documentation

- 4.71.2.1 void catcher ( int *sig* )
- 4.71.2.2 void checked\_free ( void \* *buffer* )
- 4.71.2.3 void\* checked\_malloc ( int *size* )
- 4.71.2.4 int compare\_ints ( const void \* *a*, const void \* *b* )
- 4.71.2.5 void dump\_trace ( void )
- 4.71.2.6 void error ( const char \* *a* )
- 4.71.2.7 int get\_double ( char \* *line*, int *i*, double \* *num* )
- 4.71.2.8 int get\_int ( char \* *line*, int *i*, int \* *num* )
- 4.71.2.9 double\* get\_profile\_times ( void )
- 4.71.2.10 int get\_string ( char \* *line*, int *i*, char \*\* *str* )
- 4.71.2.11 int max ( int *a*, int *b* )
- 4.71.2.12 void print\_allocs ( void )
- 4.71.2.13 void print\_profile ( void )
- 4.71.2.14 void print\_trace ( void )
- 4.71.2.15 void profile\_crash\_all\_end ( void )
- 4.71.2.16 void profile\_time\_end ( int *id* )
- 4.71.2.17 void profile\_time\_start ( int *id*, const char \* *desc* )
- 4.71.2.18 void reset\_vars ( void )
- 4.71.2.19 int set\_var ( int *var* )
- 4.71.2.20 int skip\_char ( char \* *line*, int *i*, char *c* )
- 4.71.2.21 int skip\_space ( char \* *line*, int *i* )
- 4.71.2.22 int skip\_to\_char ( char \* *line*, int *i*, char *c* )
- 4.71.2.23 char\* str\_cat ( char \* *a*, const char \* *b* )
- 4.71.2.24 char\* str\_cat\_recursive ( char \* *a*, char \* *b*, const char \* *s* )
- 4.71.2.25 char\* str\_cat\_sep ( char \* *a*, char \* *b*, char \* *s* )
- 4.71.2.26 char\* str\_copy ( char \* *a* )
- 4.71.2.27 char\* str\_copy\_short ( char \* *a*, int *max* )

4.71.2.28 `int string_equal ( char * a, char * b )`

4.71.2.29 `int string_equal_casted ( void * a, void * b )`

4.71.2.30 `void unvisit_var ( int var )`

4.71.2.31 `int visit_var ( int var )`

4.71.2.32 `void warn ( const char * a )`

## 4.72 variables.c File Reference

```
#include <stdio.h>
#include "variables.h"
#include "utils.h"
#include "llist.h"
#include "domains.h"
#include "rb_tree.h"
```

### Functions

- `var_val * new_var_val` (void)
- `variables * init_variables` (int count)
- `int set_value` (`variables *vars`, int var, int val)
- `void add_variable` (`llist *domain`, `variables *vars`, int id, const char \*name)
- `int no_solutions` (`variables *vars`)
- `void print_assigned_vars` (`variables *vars`)
- `void print_solution` (`variables *vars`)
- `int compare_variable_number` (void \*value, void \*bodyb)
- `int get_variable_number` (void \*var, void \*helper)
- `void print_variables` (`variables *vars`, const char \*notes, int dom)
- `void print_vars_for_sol_checker` (`variables *vars`, char \*name)
- `void set_static_variable_ordering` (`variables *vars`, char \*filename)
- `void init_domain_sets` (`variables *vars`)
- `void print_variable_domain_size` (`variables *vars`, int id)
- `void reset_variable_constraints` (`variables *vars`)
- `int scope_contained_in` (int \*scopea, `llist *scopeal`, int \*scopeb, `llist *scopebl`, int \*varmap, `set *projected_on`)

*if b is inside a then 1*

### 4.72.1 Function Documentation

4.72.1.1 `void add_variable ( llist * domain, variables * vars, int id, const char * name )`

4.72.1.2 `int compare_variable_number ( void * value, void * bodyb )`

4.72.1.3 `int get_variable_number ( void * var, void * helper )`

4.72.1.4 `void init_domain_sets ( variables * vars )`

4.72.1.5 `variables* init_variables ( int count )`



- 4.72.1.6 `var_val* new_var_val ( void )`
  - 4.72.1.7 `int no_solutions ( variables * vars )`
  - 4.72.1.8 `void print_assigned_vars ( variables * vars )`
  - 4.72.1.9 `void print_solution ( variables * vars )`
  - 4.72.1.10 `void print_variable_domain_size ( variables * vars, int id )`
  - 4.72.1.11 `void print_variables ( variables * vars, const char * notes, int dom )`
  - 4.72.1.12 `void print_vars_for_sol_checker ( variables * vars, char * name )`
  - 4.72.1.13 `void reset_variable_constraints ( variables * vars )`
  - 4.72.1.14 `int scope_contained_in ( int * scopea, llist * scopeal, int * scopeb, llist * scopebl, int * varmap, set * projected_on )`
- if b is inside a then 1
- 4.72.1.15 `void set_static_variable_ordering ( variables * vars, char * filename )`
  - 4.72.1.16 `int set_value ( variables * vars, int var, int val )`

## 4.73 variables.h File Reference

```
#include <stdlib.h>
#include "llist.h"
#include "lstack.h"
#include "domains.h"
#include "set.h"
```

### Classes

- struct [variable](#)
- struct [variables](#)
- struct [var\\_val](#)

### Typedefs

- typedef struct [variable](#) [variable](#)
- typedef struct [variables](#) [variables](#)
- typedef struct [var\\_val](#) [var\\_val](#)

### Functions

- [var\\_val](#) \* [new\\_var\\_val](#) (void)
- int [set\\_value](#) ([variables](#) \*vars, int var, int val)
- [variables](#) \* [init\\_variables](#) (int count)
- void [add\\_variable](#) (llist \*domain, [variables](#) \*vars, int id, const char \*name)
- int [compare\\_variable\\_number](#) (void \*value, void \*bodyb)

- int `get_variable_number` (void \*var, void \*helper)
- int `no_solutions` (variables \*vars)
- void `print_variables` (variables \*vars, const char \*notes, int dom)
- void `print_solution` (variables \*vars)
- void `print_vars_for_sol_checker` (variables \*vars, char \*name)
- void `set_neighbours` (variables \*vars)
- void `init_vars` (variables \*vars)
- void `set_static_variable_ordering` (variables \*vars, char \*filename)
- void `init_domain_sets` (variables \*vars)
- void `print_variable_domain_size` (variables \*vars, int id)
- void `print_assigned_vars` (variables \*vars)
- void `reset_variable_constraints` (variables \*vars)
- int `scope_contained_in` (int \*scopea, llist \*scopeal, int \*scopeb, llist \*scopebl, int \*varmap, set \*projected\_on)

*if b is inside a then 1*

### 4.73.1 Detailed Description

Provides the data structures for representing the CSP variables and methods for constructing and manipulating them.

### 4.73.2 Typedef Documentation

4.73.2.1 typedef struct var\_val var\_val

4.73.2.2 typedef struct variable variable

4.73.2.3 typedef struct variables variables

### 4.73.3 Function Documentation

4.73.3.1 void `add_variable` ( llist \* domain, variables \* vars, int id, const char \* name )

4.73.3.2 int `compare_variable_number` ( void \* value, void \* bodyb )

4.73.3.3 int `get_variable_number` ( void \* var, void \* helper )

4.73.3.4 void `init_domain_sets` ( variables \* vars )

4.73.3.5 variables\* `init_variables` ( int count )

4.73.3.6 void `init_vars` ( variables \* vars )

4.73.3.7 var\_val\* `new_var_val` ( void )

4.73.3.8 int `no_solutions` ( variables \* vars )

4.73.3.9 void `print_assigned_vars` ( variables \* vars )

4.73.3.10 void `print_solution` ( variables \* vars )

4.73.3.11 void `print_variable_domain_size` ( variables \* vars, int id )

4.73.3.12 void `print_variables` ( variables \* vars, const char \* notes, int dom )

4.73.3.13 void print\_vars\_for\_sol\_checker ( variables \* vars, char \* name )

4.73.3.14 void reset\_variable\_constraints ( variables \* vars )

4.73.3.15 int scope\_contained\_in ( int \* scopea, llist \* scopeal, int \* scopeb, llist \* scopebl, int \* varmap, set \* projected\_on )

if b is inside a then 1

4.73.3.16 void set\_neighbours ( variables \* vars )

4.73.3.17 void set\_static\_variable\_ordering ( variables \* vars, char \* filename )

4.73.3.18 int set\_value ( variables \* vars, int var, int val )