

Variables

- const Throwable *volatile [Parma_Polyhedra_Library::abandon_expensive_computations](#)

A pointer to an exception object.

Functions for the Synthesis of Linear Rankings

- template<typename PSET >
bool [Parma_Polyhedra_Library::termination_test_MS](#) (const PSET &pset)
Termination test using an improvement of the method by Mesnard and Serebrenik [BMPZ10].
- template<typename PSET >
bool [Parma_Polyhedra_Library::termination_test_MS_2](#) (const PSET &pset_before, const PSET &pset_after)
Termination test using an improvement of the method by Mesnard and Serebrenik [BMPZ10].
- template<typename PSET >
bool [Parma_Polyhedra_Library::one_affine_ranking_function_MS](#) (const PSET &pset, Generator &mu)
Termination test with witness ranking function using an improvement of the method by Mesnard and Serebrenik [BMPZ10].
- template<typename PSET >
bool [Parma_Polyhedra_Library::one_affine_ranking_function_MS_2](#) (const PSET &pset_before, const PSET &pset_after, Generator &mu)
Termination test with witness ranking function using an improvement of the method by Mesnard and Serebrenik [BMPZ10].
- template<typename PSET >
void [Parma_Polyhedra_Library::all_affine_ranking_functions_MS](#) (const PSET &pset, C_Polyhedron &mu_space)
Termination test with ranking function space using an improvement of the method by Mesnard and Serebrenik [BMPZ10].
- template<typename PSET >
void [Parma_Polyhedra_Library::all_affine_ranking_functions_MS_2](#) (const PSET &pset_before, const PSET &pset_after, C_Polyhedron &mu_space)
Termination test with ranking function space using an improvement of the method by Mesnard and Serebrenik [BMPZ10].
- template<typename PSET >
void [Parma_Polyhedra_Library::all_affine_quasi_ranking_functions_MS](#) (const PSET &pset, C_Polyhedron &decreasing_mu_space, C_Polyhedron &bounded_mu_space)
Computes the spaces of affine quasi ranking functions using an improvement of the method by Mesnard and Serebrenik [BMPZ10].
- template<typename PSET >
void [Parma_Polyhedra_Library::all_affine_quasi_ranking_functions_MS_2](#) (const PSET &pset_before, const PSET &pset_after, C_Polyhedron &decreasing_mu_space, C_Polyhedron &bounded_mu_space)
Computes the spaces of affine quasi ranking functions using an improvement of the method by Mesnard and Serebrenik [BMPZ10].
- template<typename PSET >
bool [Parma_Polyhedra_Library::termination_test_PR](#) (const PSET &pset)
Like [termination_test_MS\(\)](#) but using the method by Podelski and Rybalchenko [BMPZ10].
- template<typename PSET >
bool [Parma_Polyhedra_Library::termination_test_PR_2](#) (const PSET &pset_before, const PSET &pset_after)



Like *termination_test_MS_2()* but using an alternative formalization of the method by Podelski and Rybalchenko [BMPZ10].

- `template<typename PSET >`
`bool Parma_Polyhedra_Library::one_affine_ranking_function_PR (const PSET &pset, Generator &mu)`

Like *one_affine_ranking_function_MS()* but using the method by Podelski and Rybalchenko [BMPZ10].

- `template<typename PSET >`
`bool Parma_Polyhedra_Library::one_affine_ranking_function_PR_2 (const PSET &pset_before, const PSET &pset_after, Generator &mu)`

Like *one_affine_ranking_function_MS_2()* but using an alternative formalization of the method by Podelski and Rybalchenko [BMPZ10].

- `template<typename PSET >`
`void Parma_Polyhedra_Library::all_affine_ranking_functions_PR (const PSET &pset, NNC_Polyhedron &mu_space)`

Like *all_affine_ranking_functions_MS()* but using the method by Podelski and Rybalchenko [BMPZ10].

- `template<typename PSET >`
`void Parma_Polyhedra_Library::all_affine_ranking_functions_PR_2 (const PSET &pset_before, const PSET &pset_after, NNC_Polyhedron &mu_space)`

Like *all_affine_ranking_functions_MS_2()* but using an alternative formalization of the method by Podelski and Rybalchenko [BMPZ10].

8.1.1 Detailed Description

The core implementation of the Parma Polyhedra Library is written in C++. See Namespace, Hierarchical and Compound indexes for additional information about each single data type.

8.1.2 Define Documentation

8.1.2.1 #define PPL_VERSION_MAJOR 0

The major number of the PPL version.

8.1.2.2 #define PPL_VERSION_MINOR 12

The minor number of the PPL version.

8.1.2.3 #define PPL_VERSION_REVISION 0

The revision number of the PPL version.

8.1.2.4 #define PPL_VERSION "0.12"

A string containing the PPL version.

Let M and m denote the numbers associated to `PPL_VERSION_MAJOR` and `PPL_VERSION_MINOR`, respectively. The format of `PPL_VERSION` is M "." m if both `PPL_VERSION_REVISION` (r) and `PPL_VERSION_BETA` (b) are zero, M "." m "pre" b if `PPL_VERSION_REVISION` is zero and `PPL_VERSION_BETA` is not zero, M "." m "." r if `PPL_VERSION_REVISION` is not zero and `PPL_VERSION_BETA` is zero, M "." m "." r "pre" b if neither `PPL_VERSION_REVISION` nor `PPL_VERSION_BETA` are zero.

