

Gauss Symbolic Library

Generated by Doxygen 1.8.17

| | |
|------------------------------------|----------|
| 1 Namespace Index | 1 |
| 1.1 Namespace List | 1 |
| 2 File Index | 3 |
| 2.1 File List | 3 |
| 3 Namespace Documentation | 5 |
| 3.1 gauss Namespace Reference | 5 |
| 3.1.1 Detailed Description | 5 |
| 3.1.2 Function Documentation | 5 |
| 3.1.2.1 toLatex() | 5 |
| 3.1.2.2 toString() | 6 |
| 4 File Documentation | 7 |
| 4.1 gauss/Gauss.hpp File Reference | 7 |
| 4.1.1 Detailed Description | 11 |
| 4.1.2 Function Documentation | 11 |
| 4.1.2.1 abs() | 11 |
| 4.1.2.2 add() | 11 |
| 4.1.2.3 addPoly() | 12 |
| 4.1.2.4 addPolyFiniteField() | 12 |
| 4.1.2.5 arccos() | 13 |
| 4.1.2.6 arccosh() | 13 |
| 4.1.2.7 arccot() | 14 |
| 4.1.2.8 arccsc() | 14 |
| 4.1.2.9 arcsec() | 15 |
| 4.1.2.10 arcsin() | 15 |
| 4.1.2.11 arctan() | 15 |
| 4.1.2.12 arctanh() | 16 |
| 4.1.2.13 coefficientPoly() | 16 |
| 4.1.2.14 cos() | 17 |
| 4.1.2.15 cosh() | 17 |
| 4.1.2.16 cot() | 18 |
| 4.1.2.17 coth() | 18 |
| 4.1.2.18 csc() | 18 |
| 4.1.2.19 csch() | 19 |
| 4.1.2.20 degreePoly() | 19 |
| 4.1.2.21 derivative() | 20 |
| 4.1.2.22 det() | 20 |
| 4.1.2.23 div() | 21 |
| 4.1.2.24 divPoly() | 21 |
| 4.1.2.25 divPolyFiniteField() | 22 |
| 4.1.2.26 eval() | 22 |

| | |
|--|----|
| 4.1.2.27 <code>exp()</code> | 23 |
| 4.1.2.28 <code>factorPoly()</code> | 23 |
| 4.1.2.29 <code>freeVariables()</code> | 23 |
| 4.1.2.30 <code>gcdPoly()</code> | 24 |
| 4.1.2.31 <code>getOperand()</code> | 24 |
| 4.1.2.32 <code>identity()</code> | 25 |
| 4.1.2.33 <code>intFromLong()</code> | 25 |
| 4.1.2.34 <code>intFromString()</code> | 26 |
| 4.1.2.35 <code>inverse()</code> | 26 |
| 4.1.2.36 <code>is()</code> | 27 |
| 4.1.2.37 <code>isEqual()</code> | 27 |
| 4.1.2.38 <code>kindOf()</code> | 28 |
| 4.1.2.39 <code>lcmPoly()</code> | 28 |
| 4.1.2.40 <code>leadingCoefficientPoly()</code> | 28 |
| 4.1.2.41 <code>ln()</code> | 29 |
| 4.1.2.42 <code>log()</code> | 29 |
| 4.1.2.43 <code>matrix()</code> | 30 |
| 4.1.2.44 <code>matrixGet()</code> | 30 |
| 4.1.2.45 <code>matrixSet()</code> | 31 |
| 4.1.2.46 <code>mul()</code> | 31 |
| 4.1.2.47 <code>mulPoly()</code> | 32 |
| 4.1.2.48 <code>mulPolyFiniteField()</code> | 32 |
| 4.1.2.49 <code>numberFromDouble()</code> | 33 |
| 4.1.2.50 <code>numberFromString()</code> | 33 |
| 4.1.2.51 <code>pow()</code> | 34 |
| 4.1.2.52 <code>prime()</code> | 34 |
| 4.1.2.53 <code>primeFactors()</code> | 35 |
| 4.1.2.54 <code>projectPolyFiniteField()</code> | 35 |
| 4.1.2.55 <code>quoPoly()</code> | 36 |
| 4.1.2.56 <code>quoPolyFiniteField()</code> | 36 |
| 4.1.2.57 <code>remPoly()</code> | 37 |
| 4.1.2.58 <code>remPolyFiniteField()</code> | 37 |
| 4.1.2.59 <code>replace()</code> | 38 |
| 4.1.2.60 <code>resultantOfPoly()</code> | 38 |
| 4.1.2.61 <code>root()</code> | 39 |
| 4.1.2.62 <code>rootsOfPoly()</code> | 39 |
| 4.1.2.63 <code>sec()</code> | 39 |
| 4.1.2.64 <code>sech()</code> | 40 |
| 4.1.2.65 <code>setOperand()</code> | 40 |
| 4.1.2.66 <code>sin()</code> | 41 |
| 4.1.2.67 <code>sinh()</code> | 41 |
| 4.1.2.68 <code>solveLinear()</code> | 42 |

| | |
|--|-----------|
| 4.1.2.69 sqrt() | 42 |
| 4.1.2.70 sub() | 42 |
| 4.1.2.71 subPoly() | 43 |
| 4.1.2.72 subPolyFiniteField() | 43 |
| 4.1.2.73 svd() | 44 |
| 4.1.2.74 symbol() | 44 |
| 4.1.2.75 tan() | 45 |
| 4.1.2.76 tanh() | 45 |
| 4.1.2.77 transpose() | 46 |
| 5 Example Documentation | 47 |
| 5.1 /home/marcos/workspace/Gauss/gauss/Gauss.hpp | 47 |
| 5.2 expand | 48 |
| 5.3 reduce | 49 |
| Index | 51 |

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

| | |
|---------------------------------|---|
| gauss | 5 |
|---------------------------------|---|

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

| | |
|--|-------------------|
| gauss/ Gauss.hpp | |
| Header of the main API of this library | 7 |

Chapter 3

Namespace Documentation

3.1 gauss Namespace Reference

Typedefs

- typedef alg::expr **expr**
- typedef alg::kind **kind**

Functions

- std::string [toString](#) (expr a)
Return a string corresponding to a given expression.
- std::string [toLatex](#) (expr a, bool print_as_fractions, unsigned long max_den)
Construct a latex representation of a given expression.

3.1.1 Detailed Description

TODO: add list support. TODO: add set support.

3.1.2 Function Documentation

3.1.2.1 toLatex()

```
std::string gauss::toLatex (  
    expr a,  
    bool print_as_fractions,  
    unsigned long max_den )
```

Construct a latex representation of a given expression.

Parameters

| | | |
|----|------------------------|---|
| in | <i>a</i> | A expression. |
| in | <i>useFractions</i> | If true, print rational numbers as fractions. |
| in | <i>maxDenominators</i> | This is the maximum denominator for a fraction representing a number between [0, 1], bigger the number, bigger the precision on representing double precision floating points. Because of the nature of floating arithmetic, you may not always want this number as big as it can be. |

Returns

A string representing a expression on latex format.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

3.1.2.2 toString()

```
std::string gauss::toString (
    expr a )
```

Return a string corresponding to a given expression.

Parameters

| | |
|----------|-------------------------|
| <i>a</i> | A algebraic expression. |
|----------|-------------------------|

Returns

A human friendly string representation of a given expression.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

Chapter 4

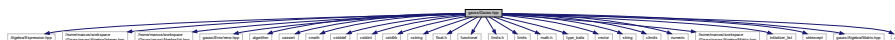
File Documentation

4.1 gauss/Gauss.hpp File Reference

Header of the main API of this library.

```
#include "Algebra/Expression.hpp"
#include "gauss/Algebra/Matrix.hpp"
#include <array>
```

Include dependency graph for Gauss.hpp:



Namespaces

- [gauss](#)

Typedefs

- typedef alg::expr **gauss::expr**
- typedef alg::kind **gauss::kind**

Functions

- expr [gauss::algebra::numberFromDouble](#) (double v)
Create a number expression from a double type.
- expr [gauss::algebra::numberFromString](#) (std::string v)
Create a number expression from a string.
- expr [gauss::algebra::intFromString](#) (const char *v)
Create a number expression from a C string value.
- expr [gauss::algebra::intFromLong](#) (long v)
Create a number expression from a long value.
- expr [gauss::algebra::symbol](#) (std::string v)
Creates a symbol expression.

- expr `gauss::algebra::pow` (expr a, expr e)
Create a power expression a^e .
- expr & `gauss::algebra::getOperand` (expr a, size_t i)
Return the i'th operand of a.
- void `gauss::algebra::setOperand` (expr &a, size_t i, expr b)
Set the i'th operand of a to b.
- kind `gauss::algebra::kindOf` (expr a)
Return the kind of a expression.
- bool `gauss::algebra::is` (expr a, int k)
Verify if a expression is of one of the given types.
- expr `gauss::algebra::root` (expr a, expr b)
A expression of kind root.
- expr `gauss::algebra::sqrt` (expr a)
A square root expression.
- bool `gauss::algebra::isEqual` (expr a, expr b)
Compute if two expressions are equal.
- expr `gauss::algebra::powDegree` (expr a)
- expr `gauss::algebra::powBase` (expr a)
- expr `gauss::algebra::rootIndex` (expr a)
- expr `gauss::algebra::rootRadicand` (expr a)
- expr `gauss::algebra::numerator` (expr a)
- expr `gauss::algebra::denominator` (expr b)
- expr `gauss::algebra::add` (expr a, expr b)
Creates an expression of form $a + b$.
- expr `gauss::algebra::sub` (expr a, expr b)
Creates an expression of form $a - b$.
- expr `gauss::algebra::mul` (expr a, expr b)
*Creates an expression of form $a * b$.*
- expr `gauss::algebra::div` (expr a, expr b)
Creates an expression of form $\begin{equation} \frac{a}{b} \end{equation}$. $x = 2$.
- expr `gauss::algebra::expand` (expr a)
- expr `gauss::algebra::reduce` (expr a)
- expr `gauss::algebra::log` (expr x, expr base)
Return a expression corresponding to a call of the logarithmic function on 'x' with a given base.
- expr `gauss::algebra::exp` (expr x)
Return a expression corresponding to a call of the exponential function on 'x'.
- expr `gauss::algebra::abs` (expr x)
Return a expression corresponding to a call of the absolute function on 'x'.
- expr `gauss::algebra::ln` (expr x)
Return a expression corresponding to a call of the natural logarithmic function on 'x'.
- expr `gauss::algebra::replace` (expr u, expr x, expr v)
Replaces x on u by v.
- expr `gauss::algebra::eval` (expr u, expr x, expr v)
Replaces x on u by v and expand the resulting expression.
- expr `gauss::algebra::freeVariables` (expr u)
Return all free variables of the expression.
- expr `gauss::algebra::prime` (size_t i)
Get the i'th prime number.
- expr `gauss::algebra::primeFactors` (expr a)
Compute the unique prime factorization of an integer.
- expr `gauss::algebra::trigonometry::sinh` (expr x)

- Return a expression corresponding to a call of the hyperbolic sine function on 'x'.*

 - expr [gauss::algebra::trigonometry::cosh](#) (expr x)
- Return a expression corresponding to a call of the hyperbolic cosine function on 'x'.*

 - expr [gauss::algebra::trigonometry::tanh](#) (expr x)
- Return a expression corresponding to a call of the hyperbolic tangent function on 'x'.*

 - expr [gauss::algebra::trigonometry::cos](#) (expr x)
- Return a expression corresponding to a call of the cosine function on 'x'.*

 - expr [gauss::algebra::trigonometry::sin](#) (expr x)
- Return a expression corresponding to a call of the sine function on 'x'.*

 - expr [gauss::algebra::trigonometry::tan](#) (expr x)
- Return a expression corresponding to a call of the tangent function on 'x'.*

 - expr [gauss::algebra::trigonometry::csc](#) (expr x)
- Return a expression corresponding to a call of the cosecant function on 'x'.*

 - expr [gauss::algebra::trigonometry::cot](#) (expr x)
- Return a expression corresponding to a call of the cotangent function on 'x'.*

 - expr [gauss::algebra::trigonometry::sec](#) (expr x)
- Return a expression corresponding to a call of the secant function on 'x'.*

 - expr [gauss::algebra::trigonometry::coth](#) (expr x)
- Return a expression corresponding to a call of the hyperbolic cotangent function on 'x'.*

 - expr [gauss::algebra::trigonometry::sech](#) (expr x)
- Return a expression corresponding to a call of the hyperbolic secant function on 'x'.*

 - expr [gauss::algebra::trigonometry::csch](#) (expr x)
- Return a expression corresponding to a call of the hyperbolic cosecant function on 'x'.*

 - expr [gauss::algebra::trigonometry::arccos](#) (expr x)
- Return a expression corresponding to a call of the arc cosine function on 'x'.*

 - expr [gauss::algebra::trigonometry::arcsin](#) (expr x)
- Return a expression corresponding to a call of the arc sine function on 'x'.*

 - expr [gauss::algebra::trigonometry::arctan](#) (expr x)
- Return a expression corresponding to a call of the arc tangent function on 'x'.*

 - expr [gauss::algebra::trigonometry::arccot](#) (expr x)
- Return a expression corresponding to a call of the arc cotangent function on 'x'.*

 - expr [gauss::algebra::trigonometry::arcsec](#) (expr x)
- Return a expression corresponding to a call of the arc secant function on 'x'.*

 - expr [gauss::algebra::trigonometry::arccsc](#) (expr x)
- Return a expression corresponding to a call of the arc cosecant function on 'x'.*

 - expr [gauss::algebra::trigonometry::arccosh](#) (expr x)
- Return a expression corresponding to a call of the arc hyperbolic cosecant function on 'x'.*

 - expr [gauss::algebra::trigonometry::arctanh](#) (expr x)
- Return a expression corresponding to a call of the arc hyperbolic tangent function on 'x'.*

 - expr [gauss::algebra::linear::matrix](#) (unsigned l, unsigned c)
- Creates a zero matrix.*

 - expr [gauss::algebra::linear::identity](#) (unsigned l, unsigned c)
- Creates a identity matrix.*

 - expr [gauss::algebra::linear::matrixGet](#) (expr A, unsigned i, unsigned j)
- Get a element of the matrix.*

 - void [gauss::algebra::linear::matrixSet](#) (expr A, unsigned i, unsigned j, double a)
- Set a element of the matrix.*

 - expr [gauss::algebra::linear::svd](#) (expr A)
- Computes the singular value decomposition of a matrix;.*

 - expr [gauss::algebra::linear::inverse](#) (expr A)
- Return the inverse of a given matrix.*

- expr `gauss::algebra::linear::det` (expr A)
Computes the determinant of a matrix.
- expr `gauss::algebra::linear::transpose` (expr A)
Computes the transposed form of a matrix.
- expr `gauss::algebra::linear::solveLinear` (expr A, expr b)
Solve a linear system $A \cdot x = b$.
- expr `gauss::polynomial::degreePoly` (expr f, expr x)
Return the greatest degree of f on x;.
- expr `gauss::polynomial::coefficientPoly` (expr f, expr x, expr d)
Return the coefficient of f on x^d ;
- expr `gauss::polynomial::leadingCoefficientPoly` (expr f, expr x)
Return the greatest coefficient of f on x;.
- expr `gauss::polynomial::rootsOfPoly` (expr a)
Computes the roots of a univariate polynomial.
- expr `gauss::polynomial::factorPoly` (expr poly)
Computes the content and the factors of a Multivariate Polynomial.
- expr `gauss::polynomial::resultantOfPoly` (expr a, expr b)
Computes the resultant of two Polynomials.
- expr `gauss::polynomial::addPoly` (expr a, expr b)
Add two polynomials.
- expr `gauss::polynomial::subPoly` (expr a, expr b)
Subtract two polynomials.
- expr `gauss::polynomial::mulPoly` (expr a, expr b)
Multiply two polynomials.
- expr `gauss::polynomial::divPoly` (expr a, expr b)
Divide two polynomial expressions.
- expr `gauss::polynomial::quoPoly` (expr a, expr b)
Compute the quotient of the polynomial division of a and b.
- expr `gauss::polynomial::remPoly` (expr a, expr b)
Compute the remainder of the polynomial division of a and b.
- expr `gauss::polynomial::gcdPoly` (expr a, expr b)
Compute the greatest common divisor of two polynomials.
- expr `gauss::polynomial::lcmPoly` (expr a, expr b)
Compute the least common multiple of two polynomials.
- expr `gauss::polynomial::finiteField::projectPolyFiniteField` (expr a, long long p)
Compute a mod p;.
- expr `gauss::polynomial::finiteField::addPolyFiniteField` (expr a, expr b, long long p)
Add two polynomial on the finite field of length 'p';.
- expr `gauss::polynomial::finiteField::subPolyFiniteField` (expr a, expr b, long long p)
Subtract two polynomial on the finite field of length 'p';.
- expr `gauss::polynomial::finiteField::mulPolyFiniteField` (expr a, expr b, long long p)
Multiply two polynomial on the finite field of length 'p';.
- expr `gauss::polynomial::finiteField::divPolyFiniteField` (expr a, expr b, long long p)
Divide two polynomial on the finite field of length 'p';.
- expr `gauss::polynomial::finiteField::quoPolyFiniteField` (expr a, expr b, long long p)
Compute the quotient of a/b on the finite field of length 'p';.
- expr `gauss::polynomial::finiteField::remPolyFiniteField` (expr a, expr b, long long p)
Compute the remainder of a/b on the finite field of length 'p';.
- expr `gauss::calculus::derivative` (expr a, expr x)
Compute the derivative of a algebraic expression.
- std::string `gauss::toString` (expr a)
Return a string corresponding to a given expression.
- std::string `gauss::toLatex` (expr a, bool print_as_fractions, unsigned long max_den)
Construct a latex representation of a given expression.

4.1.1 Detailed Description

Header of the main API of this library.

Date

Sun Mar 6 20:15:51 2022

Author

Marcos Vinicius Moreira Santos.

Copyright

BSD-3-Clause

This module defines all the public API methods of the Gauss library.

4.1.2 Function Documentation

4.1.2.1 abs()

```
expr gauss::algebra::abs (  
    expr x )
```

Return a expression corresponding to a call of the absolute function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the absolute function on x.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.2 add()

```
expr gauss::algebra::add (  
    expr a,  
    expr b )
```

Creates an expression of form $a + b$.

Creates an expression of form $a + b$, this function does not evaluate the addition, the result can be computed by a reduction, that is 'reduce(add(a, b))'.

Parameters

| | | |
|----|-----|--------------------------|
| in | a | An algebraic expression. |
| in | b | An algebraic expression. |

Returns

A new expression with the form $a + b$.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.3 addPoly()

```
expr gauss::polynomial::addPoly (
    expr a,
    expr b )
```

Add two polynomials.

Parameters

| | | |
|----|-----|--------------------------|
| in | a | A polynomial expression. |
| in | b | A polynomial expression. |

Returns

The polynomial resulting of the addition of a and b .

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.4 addPolyFiniteField()

```
expr gauss::polynomial::finiteField::addPolyFiniteField (
    expr a,
    expr b,
    long long p )
```

Add two polynomial on the finite field of length 'p';.

Parameters

| | | |
|----|-----|-------------------------|
| in | a | A polynomial expression |
| in | b | A polynomial expression |
| in | p | A integer. |

Returns

$(a + b) \bmod p$

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.5 arccos()

```
expr gauss::algebra::trigonometry::arccos (
    expr x )
```

Return a expression corresponding to a call of the arc cosine function on 'x'.

Parameters

| | | |
|----|-----|----------------|
| in | x | An expression. |
|----|-----|----------------|

Returns

a call to the arc cosine function on x.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.6 arccosh()

```
expr gauss::algebra::trigonometry::arccosh (
    expr x )
```

Return a expression corresponding to a call of the arc hyperbolic cosecant function on 'x'.

Parameters

| | | |
|----|-----|----------------|
| in | x | An expression. |
|----|-----|----------------|

Returns

a call to the arc hyperbolic cosecant function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.7 arccot()

```
expr gauss::algebra::trigonometry::arccot (
    expr x )
```

Return a expression corresponding to a call of the arc cotangent function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the arc cotangent function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.8 arccsc()

```
expr gauss::algebra::trigonometry::arccsc (
    expr x )
```

Return a expression corresponding to a call of the arc cosecant function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the arc cosecant function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.9 arcsec()

```
expr gauss::algebra::trigonometry::arcsec (  
    expr x )
```

Return a expression corresponding to a call of the arc secant function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the arc secant function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.10 arcsin()

```
expr gauss::algebra::trigonometry::arcsin (  
    expr x )
```

Return a expression corresponding to a call of the arc sine function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the arc sine function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.11 arctan()

```
expr gauss::algebra::trigonometry::arctan (  
    expr x )
```

Return a expression corresponding to a call of the arc tangent function on 'x'.

Parameters

| | | |
|----|-----|----------------|
| in | x | An expression. |
|----|-----|----------------|

Returns

a call to the arc tangent function on x .

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.12 arctanh()

```
expr gauss::algebra::trigonometry::arctanh (
    expr x )
```

Return a expression corresponding to a call of the arc hyperbolic tangent function on ' x '.

Parameters

| | | |
|----|-----|----------------|
| in | x | An expression. |
|----|-----|----------------|

Returns

a call to the arc hyperbolic tangent function on x .

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.13 coefficientPoly()

```
expr gauss::polynomial::coefficientPoly (
    expr f,
    expr x,
    expr d )
```

Return the coefficient of f on x^d .

Parameters

| | | |
|----|-----|---------------|
| in | f | A expression. |
| in | x | A symbol. |
| in | d | A integer. |

Returns

The coefficient of f on x^d .

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.14 cos()

```
expr gauss::algebra::trigonometry::cos (  
    expr x )
```

Return a expression corresponding to a call of the cosine function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the cosine function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.15 cosh()

```
expr gauss::algebra::trigonometry::cosh (  
    expr x )
```

Return a expression corresponding to a call of the hyperbolic cosine function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the hyperbolic cosine function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.16 cot()

```
expr gauss::algebra::trigonometry::cot (  
    expr x )
```

Return a expression corresponding to a call of the cotangent function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the cotangent function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.17 coth()

```
expr gauss::algebra::trigonometry::coth (  
    expr x )
```

Return a expression corresponding to a call of the hyperbolic cotangent function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the hyperbolic cotangent function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.18 csc()

```
expr gauss::algebra::trigonometry::csc (  
    expr x )
```

Return a expression corresponding to a call of the cosecant function on 'x'.

Parameters

| | | |
|----|-----|----------------|
| in | x | An expression. |
|----|-----|----------------|

Returns

a call to the cosecant function on x .

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.19 csch()

```
expr gauss::algebra::trigonometry::csch (
    expr x )
```

Return a expression corresponding to a call of the hyperbolic cosecant function on ' x '.

Parameters

| | | |
|----|-----|----------------|
| in | x | An expression. |
|----|-----|----------------|

Returns

a call to the hyperbolic cosecant function on x .

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.20 degreePoly()

```
expr gauss::polynomial::degreePoly (
    expr f,
    expr x )
```

Return the greatest degree of f on x ;

Parameters

| | | |
|----|-----|---------------|
| in | f | A expression. |
| in | x | A symbol. |

Returns

The degree greatest of f on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.21 derivative()

```
expr gauss::calculus::derivative (  
    expr a,  
    expr x )
```

Compute the derivative of a algebraic expression.

Computes the derivative of the expression 'a' on the variable 'x' using elementary calculus methods.

Parameters

| | | |
|----|---|---------------------------------------|
| in | a | An algebraic expression |
| in | x | A free variable of the expression 'a' |

Returns

The algebraic expression corresponding to the derivative of 'a' by 'x'

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.22 det()

```
expr gauss::algebra::linear::det (  
    expr A )
```

Computes the determinant of a matrix.

Parameters

| | | |
|----|---|------------------------|
| in | A | The matrix expression. |
|----|---|------------------------|

Returns

The determinant of the matrix.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.23 div()

```
expr gauss::algebra::div (
    expr a,
    expr b )
```

Creates an expression of form $\frac{a}{b}$. $x = 2$.

Creates an expression of form a / b , this function does not evaluate the subtraction, the result can be computed by a reduction, that is 'reduce(div(a, b))'.

Parameters

| | | |
|----|----------|--------------------------|
| in | <i>a</i> | An algebraic expression. |
| in | <i>b</i> | An algebraic expression. |

Returns

A new expression with the form a / b .

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.24 divPoly()

```
expr gauss::polynomial::divPoly (
    expr a,
    expr b )
```

Divide two polynomial expressions.

Parameters

| | | |
|----|----------|--------------------------|
| in | <i>a</i> | A polynomial expression. |
| in | <i>b</i> | A polynomial expression. |

Returns

The expression $\text{quotient}(a, b) + \text{remainder}(a, b)$

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.25 divPolyFiniteField()

```
expr gauss::polynomial::finiteField::divPolyFiniteField (
    expr a,
    expr b,
    long long p )
```

Divide two polynomial on the finite field of length 'p';.

Parameters

| | | |
|----|----------|-------------------------|
| in | <i>a</i> | A polynomial expression |
| in | <i>b</i> | A polynomial expression |
| in | <i>p</i> | A integer. |

Returns

$(a / b) \bmod p$

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.26 eval()

```
expr gauss::algebra::eval (
    expr u,
    expr x,
    expr v )
```

Replaces x on u by v and expand the resulting expression.

Parameters

| | | |
|----|----------|---------------|
| in | <i>u</i> | A expression. |
| in | <i>x</i> | A symbol. |
| in | <i>v</i> | A expression. |

Returns

A new expression without x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.27 `exp()`

```
expr gauss::algebra::exp (
    expr x )
```

Return a expression corresponding to a call of the exponential function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the exponential function on x.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.28 `factorPoly()`

```
expr gauss::polynomial::factorPoly (
    expr poly )
```

Computes the the content and the factors of a Multivariate Polynomial.

Parameters

| | | |
|----|------|-------------------------|
| in | poly | A polynomial expression |
|----|------|-------------------------|

Returns

The factorized form of a polynomial expression.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.29 `freeVariables()`

```
expr gauss::algebra::freeVariables (
    expr u )
```

Return all free variables of the expression.

Parameters

| | | |
|----|-----|---------------|
| in | u | A expression. |
|----|-----|---------------|

Returns

A set of symbols.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.30 gcdPoly()

```
expr gauss::polynomial::gcdPoly (
    expr a,
    expr b )
```

Compute the greatest common divisor of two polynomials.

Parameters

| | | |
|----|-----|--------------------------|
| in | a | A polynomial expression. |
| in | b | A polynomial expression. |

Returns

The greatest common divisor of 'a' and 'b'.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.31 getOperand()

```
expr& gauss::algebra::getOperand (
    expr a,
    size_t i )
```

Return the i th operand of a.

Parameters

| | | |
|----|-----|--------------------------------------|
| in | a | A expression with multiple operands. |
| in | i | A positive integer number. |

Returns

The i'th operand of a.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.32 identity()

```
expr gauss::algebra::linear::identity (
    unsigned l,
    unsigned c )
```

Creates a identity matrix.

Parameters

| | | |
|----|----------|----------------------------------|
| in | <i>l</i> | Number of rows of the matrix. |
| in | <i>c</i> | Number of columns of the matrix. |

Returns

A identity matrix.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.33 intFromLong()

```
expr gauss::algebra::intFromLong (
    long v )
```

Create a number expression from a long value.

Create a number expression from a long value.

Parameters

| | | |
|----|----------|-------------|
| in | <i>v</i> | long value. |
|----|----------|-------------|

Returns

A Integer expression.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.34 intFromString()

```
expr gauss::algebra::intFromString (
    const char * v )
```

Create a number expression from a C string value.

Create a number expression from a C string value. The string should represent a Integer.

Parameters

| | | |
|----|---|--|
| in | v | A C string corresponding to a Integer. |
|----|---|--|

Returns

A Integer expression.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.35 inverse()

```
expr gauss::algebra::linear::inverse (
    expr A )
```

Return the inverse of a given matrix.

Parameters

| | | |
|----|---|------------------------|
| in | A | The matrix expression. |
|----|---|------------------------|

Returns

The inverse of the matrix A.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.36 is()

```
bool gauss::algebra::is (
    expr a,
    int k )
```

Verify if a expression is of one of the given types.

Parameters

| | | |
|----|----------|---|
| in | <i>a</i> | A expression. |
| in | <i>k</i> | A integer that can be constructed from kinds with bitwise 'or'. ex: kind::INT kind::FRAC. |

Returns

True if the expression is of one of the given kinds, False otherwise.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.37 isEqual()

```
bool gauss::algebra::isEqual (
    expr a,
    expr b )
```

Compute if two expressions are equal.

Compute if the result of reduce(expand(a) - expand(b)) is equal to 0.

Parameters

| | | |
|----|----------|----------------|
| in | <i>a</i> | An expression. |
| in | <i>b</i> | An expression. |

Returns

true if reduce(expand(a) - expand(b)) == 0, false otherwise.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.38 kindOf()

```
kind gauss::algebra::kindOf (
    expr a )
```

Return the kind of a expression.

Parameters

| | | |
|----|----------|---------------|
| in | <i>a</i> | A expression. |
|----|----------|---------------|

Returns

The kind of the expression.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.39 lcmPoly()

```
expr gauss::polynomial::lcmPoly (
    expr a,
    expr b )
```

Compute the least commom multiple of two polynomials.

Parameters

| | | |
|----|----------|--------------------------|
| in | <i>a</i> | A polynomial expression. |
| in | <i>b</i> | A polynomial expression. |

Returns

The least commom multiple of 'a' and 'b'.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.40 leadingCoefficientPoly()

```
expr gauss::polynomial::leadingCoefficientPoly (
    expr f,
    expr x )
```

Return the greatest coefficient of f on x;.

Parameters

| | | |
|----|----------|---------------|
| in | <i>f</i> | A expression. |
| in | <i>x</i> | A symbol. |

Returns

The greatest coefficient of *f* on *x*.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.41 ln()

```
expr gauss::algebra::ln (
    expr x )
```

Return a expression corresponding to a call of the natural logarithmic function on 'x'.

Parameters

| | | |
|----|----------|----------------|
| in | <i>x</i> | An expression. |
|----|----------|----------------|

Returns

a call to the natural logarithmic function on *x*.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.42 log()

```
expr gauss::algebra::log (
    expr x,
    expr base )
```

Return a expression corresponding to a call of the logarithmic function on 'x' with a given base.

Parameters

| | | |
|----|-------------|----------------|
| in | <i>x</i> | An expression. |
| in | <i>base</i> | An expression. |

Returns

a call to the logarithmic function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.43 matrix()

```
expr gauss::algebra::linear::matrix (
    unsigned l,
    unsigned c )
```

Creates a zero matrix.

Parameters

| | | |
|----|----------|----------------------------------|
| in | <i>l</i> | Number of rows of the matrix. |
| in | <i>c</i> | Number of columns of the matrix. |

Returns

A matrix filled with zeros.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.44 matrixGet()

```
expr gauss::algebra::linear::matrixGet (
    expr A,
    unsigned i,
    unsigned j )
```

Get a element of the matrix.

Parameters

| | | |
|----|----------|----------------------------|
| in | <i>A</i> | The matrix expression. |
| in | <i>i</i> | The row of the element. |
| in | <i>j</i> | The column of the element. |

Returns

A number in the $A[i][j]$ position;

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.45 matrixSet()

```
void gauss::algebra::linear::matrixSet (
    expr A,
    unsigned i,
    unsigned j,
    double a )
```

Set a element of the matrix.

Parameters

| | | |
|----|-----|------------------------|
| in | A | The matrix expression. |
| in | i | The row. |
| in | j | The column. |
| in | a | A double value. |

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.46 mul()

```
expr gauss::algebra::mul (
    expr a,
    expr b )
```

Creates an expression of form $a * b$;

Creates an expression of form $a * b$, this function does not evaluate the subtraction, the result can be computed by a reduction, that is 'reduce(mul(a, b))'.

Parameters

| | | |
|----|-----|--------------------------|
| in | a | An algebraic expression. |
| in | b | An algebraic expression. |

Returns

A new expression with the form $a * b$.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.47 mulPoly()

```
expr gauss::polynomial::mulPoly (
    expr a,
    expr b )
```

Multiply two polynomials.

Parameters

| | | |
|----|-----|--------------------------|
| in | a | A polynomial expression. |
| in | b | A polynomial expression. |

Returns

The polynomial resulting of the multiplication of a and b .

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.48 mulPolyFiniteField()

```
expr gauss::polynomial::finiteField::mulPolyFiniteField (
    expr a,
    expr b,
    long long p )
```

Multiply two polynomial on the finite field of length ' p '.

Parameters

| | | |
|----|-----|-------------------------|
| in | a | A polynomial expression |
| in | b | A polynomial expression |
| in | p | A integer. |

Returns $(a * b) \bmod p$ **Examples**</home/marcos/workspace/Gauss/ gauss/ Gauss. hpp.>**4.1.2.49 numberFromDouble()**

```

expr gauss::algebra::numberFromDouble (
    double v )

```

Create a number expression from a double type.

Create a number expression from a double type. The returned value will be a fraction if the given value has a fractional part greater than the machine epsilon or a integer if the given value does not have a fractional part.

Parameters

| | | |
|----|---|-----------------|
| in | v | A double value. |
|----|---|-----------------|

Returns

The closest fraction to the double value considering the maximum denominator specified.

Examples</home/marcos/workspace/Gauss/ gauss/ Gauss. hpp.>**4.1.2.50 numberFromString()**

```

expr gauss::algebra::numberFromString (
    std::string v )

```

Create a number expression from a string.

Create a number expression from a string type. The returned value will be a fraction if the given value has a fractional part greater than the machine epsilon or a integer if the given value does not have a fractional part.

Parameters

| | | |
|----|---|----------------------|
| in | v | A const char* value. |
|----|---|----------------------|

Returns

The closest fraction to the double value considering the maximum denominator specified.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.51 pow()

```
expr gauss::algebra::pow (
    expr a,
    expr e )
```

Create a power expression a^e .

Parameters

| | | |
|----|----------|---------------------------|
| in | <i>a</i> | The base of the power. |
| in | <i>e</i> | The expoent of the power. |

Returns

A power expression.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.52 prime()

```
expr gauss::algebra::prime (
    size_t i )
```

Get the i 'th prime number.

Computes the i 'th prime number using the sieve of Erastostenes. This function can only compute primes up to 18446744073709551615 assuming enough memory.

Parameters

| | | |
|----|----------|--|
| in | <i>i</i> | The index of the prime to be computed. |
|----|----------|--|

Returns

Return the i 'th prime number.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.53 primeFactors()

```
expr gauss::algebra::primeFactors (
    expr a )
```

Compute the unique prime factorization of an integer.

Compute the unique prime factorization of an integer and return the product of all the prime factors, if the number is less than 0, -1 is returned as the first element.

Parameters

| | | |
|----|-----|-------------------------|
| in | A | number of kind integer. |
|----|-----|-------------------------|

Returns

The product of all prime factors of a number;

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.54 projectPolyFiniteField()

```
expr gauss::polynomial::finiteField::projectPolyFiniteField (
    expr a,
    long long p )
```

Compute $a \bmod p$.

Parameters

| | | |
|----|-----|--------------------------|
| in | a | A polynomial expression. |
| in | p | A long long integer. |

Returns

$a \bmod p$

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.55 quoPoly()

```
expr gauss::polynomial::quoPoly (
    expr a,
    expr b )
```

Compute the quotient of the polynomial division of a and b.

Parameters

| | | |
|----|----------|--------------------------|
| in | <i>a</i> | A polynomial expression. |
| in | <i>b</i> | A polynomial expression. |

Returns

The expression corresponding to the quotient of 'a / b'

Examples

</home/marcos/workspace/Gauss/qauss/Gauss.hpp>.

4.1.2.56 quoPolyFiniteField()

```
expr gauss::polynomial::finiteField::quoPolyFiniteField (
    expr a,
    expr b,
    long long p )
```

Compute the quotient of a/b on the finite field of length 'p';.

Parameters

| | | |
|----|----------|-------------------------|
| in | <i>a</i> | A polynomial expression |
| in | <i>b</i> | A polynomial expression |
| in | <i>p</i> | A integer. |

Returns

quotient(a, b) mod p

Examples

</home/marcos/workspace/Gauss/qauss/Gauss.hpp>.

4.1.2.57 remPoly()

```
expr gauss::polynomial::remPoly (  
    expr a,  
    expr b )
```

Compute the remainder of the polynomial division of a and b.

Parameters

| | | |
|----|-----|--------------------------|
| in | a | A polynomial expression. |
| in | b | A polynomial expression. |

Returns

The expression corresponding to the remainder of 'a / b'

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.58 remPolyFiniteField()

```
expr gauss::polynomial::finiteField::remPolyFiniteField (  
    expr a,  
    expr b,  
    long long p )
```

Compute the remainder of a/b on the finite field of length 'p';

Parameters

| | | |
|----|-----|-------------------------|
| in | a | A polynomial expression |
| in | b | A polynomial expression |
| in | p | A integer. |

Returns

remainder(a, b) mod p

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.59 replace()

```
expr gauss::algebra::replace (
    expr u,
    expr x,
    expr v )
```

Replaces x on u by v.

Parameters

| | | |
|----|-----|---------------|
| in | u | A expression. |
| in | x | A symbol. |
| in | v | A expression. |

Returns

u with all occurrences of x replaced by v.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.60 resultantOfPoly()

```
expr gauss::polynomial::resultantOfPoly (
    expr a,
    expr b )
```

Computes the resultant of two Polynomials.

Parameters

| | | |
|----|-----|--------------------------|
| in | a | A polynomial expression. |
| in | b | A polynomial expression. |

Returns

The resultant polynomial of a and b.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.61 root()

```
expr gauss::algebra::root (
    expr a,
    expr b )
```

A expression of kind root.

Parameters

| | | |
|----|----------|-------------------------------------|
| in | <i>a</i> | The radical of the root expression. |
| in | <i>b</i> | The index of the root expression. |

Returns

A root expression.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.62 rootsOfPoly()

```
expr gauss::polynomial::rootsOfPoly (
    expr a )
```

Computes the roots of a univariate polynomial.

Computes the roots of the polynomial using the Jenkins and Traub Algorithm.

Parameters

| | | |
|----|----------|-----------------------|
| in | <i>a</i> | Univariate Polynomial |
|----|----------|-----------------------|

Returns

A list with the roots of the polynomial.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.63 sec()

```
expr gauss::algebra::trigonometry::sec (
    expr x )
```

Return a expression corresponding to a call of the secant function on 'x'.

Parameters

| | | |
|----|-----|----------------|
| in | x | An expression. |
|----|-----|----------------|

Returns

a call to the secant function on x .

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.64 sech()

```
expr gauss::algebra::trigonometry::sech (
    expr x )
```

Return a expression corresponding to a call of the hyperbolic secant function on ' x '.

Parameters

| | | |
|----|-----|----------------|
| in | x | An expression. |
|----|-----|----------------|

Returns

a call to the hyperbolic secant function on x .

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.65 setOperand()

```
void gauss::algebra::setOperand (
    expr & a,
    size_t i,
    expr b )
```

Set the i 'th operand of a to b .

Parameters

| | | |
|----|-----|-----------------------------|
| in | a | A multi operand expression. |
| in | i | A positive integer number. |
| in | b | A expression. |

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.66 sin()

```
expr gauss::algebra::trigonometry::sin (  
    expr x )
```

Return a expression corresponding to a call of the sine function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the sine function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.67 sinh()

```
expr gauss::algebra::trigonometry::sinh (  
    expr x )
```

Return a expression corresponding to a call of the hyperbolic sine function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the hyperbolic sine function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.68 solveLinear()

```
expr gauss::algebra::linear::solveLinear (
    expr A,
    expr b )
```

Solve a linear system $A*x = b$.

Parameters

| | | |
|----|-----|-------------------------|
| in | A | Matrix of coefficients. |
| in | b | Vector of solutions. |

Returns

The vector x .

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.69 sqrt()

```
expr gauss::algebra::sqrt (
    expr a )
```

A square root expression.

Parameters

| | | |
|----|-----|--------------------------------|
| in | a | The radical of the expression. |
|----|-----|--------------------------------|

Returns

A square root expression.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.70 sub()

```
expr gauss::algebra::sub (
    expr a,
    expr b )
```

Creates an expression of form $a - b$.

Creates an expression of form $a - b$, this function does not evaluate the subtraction, the result can be computed by a reduction, that is 'reduce(sub(a, b))'.

Parameters

| | | |
|----|----------|--------------------------|
| in | <i>a</i> | An algebraic expression. |
| in | <i>b</i> | An algebraic expression. |

Returns

A new expression with the form $a - b$.

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.71 subPoly()

```
expr gauss::polynomial::subPoly (
    expr a,
    expr b )
```

Subtract two polynomials.

Parameters

| | | |
|----|----------|--------------------------|
| in | <i>a</i> | A polynomial expression. |
| in | <i>b</i> | A polynomial expression. |

Returns

The polynomial resulting of the subtraction of a and b .

Examples

</home/marcos/workspace/Gauss/ gauss/ Gauss.hpp>.

4.1.2.72 subPolyFiniteField()

```
expr gauss::polynomial::finiteField::subPolyFiniteField (
    expr a,
    expr b,
    long long p )
```

Subtract two polynomial on the finite field of length ' p '.

Parameters

| | | |
|----|-----|-------------------------|
| in | a | A polynomial expression |
| in | b | A polynomial expression |
| in | p | A integer. |

Returns

$(a - b) \bmod p$

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.73 svd()

```
expr gauss::algebra::linear::svd (
    expr A )
```

Computes the singular value decomposition of a matrix;.

Parameters

| | | |
|----|-----|------------------------|
| in | A | The matrix expression. |
|----|-----|------------------------|

Returns

A list with the matrices [U, D, transpose(V)].

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.74 symbol()

```
expr gauss::algebra::symbol (
    std::string v )
```

Creates a symbol expression.

Creates a symbol expression.

Parameters

| | | |
|----|-----|--|
| in | v | A C string corresponding to the symbol identifier. |
|----|-----|--|

Returns

A symbol expression.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.75 tan()

```
expr gauss::algebra::trigonometry::tan (  
    expr x )
```

Return a expression corresponding to a call of the tangent function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the tangent function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.76 tanh()

```
expr gauss::algebra::trigonometry::tanh (  
    expr x )
```

Return a expression corresponding to a call of the hyperbolic tangent function on 'x'.

Parameters

| | | |
|----|---|----------------|
| in | x | An expression. |
|----|---|----------------|

Returns

a call to the hyperbolic tangent function on x.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

4.1.2.77 transpose()

```
expr gauss::algebra::linear::transpose (  
    expr A )
```

Computes the tranposed form of a matrix.

Parameters

| | | |
|----|----------|------------------------|
| in | <i>A</i> | The matrix expression. |
|----|----------|------------------------|

Returns

The transposed form of 'A'.

Examples

</home/marcos/workspace/Gauss/gauss/Gauss.hpp>.

Chapter 5

Example Documentation

5.1 /home/marcos/workspace/Gauss/gauss/Gauss.hpp

Return the degree expression of a power expression. Compute the degree of a power expression. if the expression is not a power, an error is raised.

Parameters

| | | |
|----|----------|---------------------|
| in | <i>a</i> | A power expression. |
|----|----------|---------------------|

```
expr x = symbol("x"); expr t = pow(x, 2); assert(isEqual(powDegree(t), 2));
```

Returns

The degree of the power expression.

```
#include "Algebra/Expression.hpp"
#include "gauss/Algebra/Matrix.hpp"
#include <array>
#include <cstdlib>
#include <string>
namespace gauss {
typedef alg::expr expr;
typedef alg::kind kind;
namespace algebra {
expr numberFromDouble(double v);
expr numberFromString(std::string v);
expr intFromString(const char *v);
expr intFromLong(long v);
expr symbol(std::string v);
expr pow(expr a, expr e);
expr &getOperand(expr a, size_t i);
void setOperand(expr &a, size_t i, expr b);
kind kindOf(expr a);
bool is(expr a, int k);
expr root(expr a, expr b);
expr sqrt(expr a);
bool isEqual(expr a, expr b);
expr powDegree(expr a);
expr powBase(expr a);
expr rootIndex(expr a);
expr rootRadicand(expr a);
expr numerator(expr a);
expr denominator(expr b);
expr add(expr a, expr b);
expr sub(expr a, expr b);
expr mul(expr a, expr b);
expr div(expr a, expr b);
expr expand(expr a);
expr reduce(expr a);
```

```

expr log(expr x, expr base);
expr exp(expr x);
expr abs(expr x);
expr ln(expr x);
expr replace(expr u, expr x, expr v);
expr eval(expr u, expr x, expr v);
expr freeVariables(expr u);
expr prime(size_t i);
expr primeFactors(expr a);
namespace trigonometry {
expr sinh(expr x);
expr cosh(expr x);
expr tanh(expr x);
expr cos(expr x);
expr sin(expr x);
expr tan(expr x);
expr csc(expr x);
expr cot(expr x);
expr sec(expr x);
expr coth(expr x);
expr sech(expr x);
expr csch(expr x);
expr arccos(expr x);
expr arcsin(expr x);
expr arctan(expr x);
expr arccot(expr x);
expr arcsec(expr x);
expr arccsc(expr x);
expr arccosh(expr x);
expr arctanh(expr x);
}; // namespace trigonometry
namespace linear {
expr matrix(unsigned l, unsigned c);
expr identity(unsigned l, unsigned c);
expr matrixGet(expr A, unsigned i, unsigned j);
void matrixSet(expr A, unsigned i, unsigned j, double a);
expr svd(expr A);
expr inverse(expr A);
expr det(expr A);
expr transpose(expr A);
expr solveLinear(expr A, expr b);
} // namespace linear
} // namespace algebra
namespace polynomial {
expr degreePoly(expr f, expr x);
expr coefficientPoly(expr f, expr x,
                     expr d);
expr leadingCoefficientPoly(expr f, expr x);
expr rootsOfPoly(expr a);
expr factorPoly(expr poly);
expr resultantOfPoly(expr a, expr b);
expr addPoly(expr a, expr b);
expr subPoly(expr a, expr b);
expr mulPoly(expr a, expr b);
expr divPoly(expr a, expr b);
expr quoPoly(expr a, expr b);
expr remPoly(expr a, expr b);
expr gcdPoly(expr a, expr b);
expr lcmPoly(expr a, expr b);
namespace finiteField {
expr projectPolyFiniteField(expr a, long long p);
expr addPolyFiniteField(expr a, expr b, long long p);
expr subPolyFiniteField(expr a, expr b, long long p);
expr mulPolyFiniteField(expr a, expr b, long long p);
expr divPolyFiniteField(expr a, expr b, long long p);
expr quoPolyFiniteField(expr a, expr b, long long p);
expr remPolyFiniteField(expr a, expr b, long long p);
} // namespace finiteField
} // namespace polynomial
namespace calculus {
expr derivative(expr a, expr x);
} // namespace calculus
std::string toString(expr a);
std::string toLatex(expr a, bool print_as_fractions,
                   unsigned long max_den);
} // namespace gauss

```

5.2 expand

Expand a expression. Expand and reduce an expression.

$(x(3x + 4)) = 3x^2 + 4x$.

Returns

A algebraic expression corresponding to the expansion of the expression 'a'.

5.3 reduce

Reduce an expression. Reduce an expression to the smallest possible form not regarding algebraic equalities or expansions. That means that it performs the operations of a given expression.

Parameters

| | | |
|----|---|--------------------------|
| in | a | An algebraic expression. |
|----|---|--------------------------|

$$(3x + 4y^2 + 5x + (3x + 3y^2)) = 11x + 7y^2$$

Returns

The reduced form of 'a'

Index

abs
 Gauss.hpp, 11
add
 Gauss.hpp, 11
addPoly
 Gauss.hpp, 12
addPolyFiniteField
 Gauss.hpp, 12
arccos
 Gauss.hpp, 13
arccosh
 Gauss.hpp, 13
arccot
 Gauss.hpp, 14
arccsc
 Gauss.hpp, 14
arcsec
 Gauss.hpp, 14
arcsin
 Gauss.hpp, 15
arctan
 Gauss.hpp, 15
arctanh
 Gauss.hpp, 16

coefficientPoly
 Gauss.hpp, 16
cos
 Gauss.hpp, 17
cosh
 Gauss.hpp, 17
cot
 Gauss.hpp, 17
coth
 Gauss.hpp, 18
csc
 Gauss.hpp, 18
csch
 Gauss.hpp, 19

degreePoly
 Gauss.hpp, 19
derivative
 Gauss.hpp, 20
det
 Gauss.hpp, 20
div
 Gauss.hpp, 21
divPoly
 Gauss.hpp, 21

divPolyFiniteField
 Gauss.hpp, 22

eval
 Gauss.hpp, 22
exp
 Gauss.hpp, 22

factorPoly
 Gauss.hpp, 23
freeVariables
 Gauss.hpp, 23

gauss, 5
 toLatex, 5
 toString, 6
Gauss.hpp
 abs, 11
 add, 11
 addPoly, 12
 addPolyFiniteField, 12
 arccos, 13
 arccosh, 13
 arccot, 14
 arccsc, 14
 arcsec, 14
 arcsin, 15
 arctan, 15
 arctanh, 16
 coefficientPoly, 16
 cos, 17
 cosh, 17
 cot, 17
 coth, 18
 csc, 18
 csch, 19
 degreePoly, 19
 derivative, 20
 det, 20
 div, 21
 divPoly, 21
 divPolyFiniteField, 22
 eval, 22
 exp, 22
 factorPoly, 23
 freeVariables, 23
 gcdPoly, 24
 getOperand, 24
 identity, 25
 intFromLong, 25

- intFromString, 26
- inverse, 26
- is, 26
- isEqual, 27
- kindOf, 27
- lcmPoly, 28
- leadingCoefficientPoly, 28
- ln, 29
- log, 29
- matrix, 30
- matrixGet, 30
- matrixSet, 31
- mul, 31
- mulPoly, 32
- mulPolyFiniteField, 32
- numberFromDouble, 33
- numberFromString, 33
- pow, 34
- prime, 34
- primeFactors, 35
- projectPolyFiniteField, 35
- quoPoly, 35
- quoPolyFiniteField, 36
- remPoly, 36
- remPolyFiniteField, 37
- replace, 37
- resultantOfPoly, 38
- root, 38
- rootsOfPoly, 39
- sec, 39
- sech, 40
- setOperand, 40
- sin, 41
- sinh, 41
- solveLinear, 41
- sqrt, 42
- sub, 42
- subPoly, 43
- subPolyFiniteField, 43
- svd, 44
- symbol, 44
- tan, 45
- tanh, 45
- transpose, 45
- gauss/Gauss.hpp, 7
- gcdPoly
 - Gauss.hpp, 24
- getOperand
 - Gauss.hpp, 24
- identity
 - Gauss.hpp, 25
- intFromLong
 - Gauss.hpp, 25
- intFromString
 - Gauss.hpp, 26
- inverse
 - Gauss.hpp, 26
- is
 - Gauss.hpp, 26
- isEqual
 - Gauss.hpp, 27
- kindOf
 - Gauss.hpp, 27
- lcmPoly
 - Gauss.hpp, 28
- leadingCoefficientPoly
 - Gauss.hpp, 28
- ln
 - Gauss.hpp, 29
- log
 - Gauss.hpp, 29
- matrix
 - Gauss.hpp, 30
- matrixGet
 - Gauss.hpp, 30
- matrixSet
 - Gauss.hpp, 31
- mul
 - Gauss.hpp, 31
- mulPoly
 - Gauss.hpp, 32
- mulPolyFiniteField
 - Gauss.hpp, 32
- numberFromDouble
 - Gauss.hpp, 33
- numberFromString
 - Gauss.hpp, 33
- pow
 - Gauss.hpp, 34
- prime
 - Gauss.hpp, 34
- primeFactors
 - Gauss.hpp, 35
- projectPolyFiniteField
 - Gauss.hpp, 35
- quoPoly
 - Gauss.hpp, 35
- quoPolyFiniteField
 - Gauss.hpp, 36
- remPoly
 - Gauss.hpp, 36
- remPolyFiniteField
 - Gauss.hpp, 37
- replace
 - Gauss.hpp, 37
- resultantOfPoly
 - Gauss.hpp, 38
- root
 - Gauss.hpp, 38
- rootsOfPoly
 - Gauss.hpp, 39

- sec
 - Gauss.hpp, [39](#)
- sech
 - Gauss.hpp, [40](#)
- setOperand
 - Gauss.hpp, [40](#)
- sin
 - Gauss.hpp, [41](#)
- sinh
 - Gauss.hpp, [41](#)
- solveLinear
 - Gauss.hpp, [41](#)
- sqrt
 - Gauss.hpp, [42](#)
- sub
 - Gauss.hpp, [42](#)
- subPoly
 - Gauss.hpp, [43](#)
- subPolyFiniteField
 - Gauss.hpp, [43](#)
- svd
 - Gauss.hpp, [44](#)
- symbol
 - Gauss.hpp, [44](#)
- tan
 - Gauss.hpp, [45](#)
- tanh
 - Gauss.hpp, [45](#)
- toLatex
 - gauss, [5](#)
- toString
 - gauss, [6](#)
- transpose
 - Gauss.hpp, [45](#)