MSDScript

Generated by Doxygen 1.9.6

Chapter 1

MSDScript

Author

Avishek

Date

02-02-2023

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

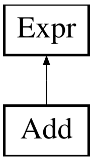
Here is a list of all documented files with brief descriptions:

Chapter 5

Class Documentation

5.1 Add Class Reference

Inheritance diagram for Add:



Public Member Functions

- Add (Expr ∗lhs, Expr ∗rhs)

    Constructor takes two arguments to create Add object.
- bool Equals (Expr ∗expr)

    Compares two Add type expressions.
- int interp ()

    Interpret the expression to an integer value.
- bool has_variable ()

    checks if the expression has a variable
- Expr ∗ subst (std::string x, Expr ∗exp)

    Substitutes a variable present in the expression with a given expression.

- virtual bool Equals (Expr ∗e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr ∗ subst (std::string x, Expr ∗exp)=0

Public Attributes

- Expr ∗ **lhs**

    left hand side of the Add expression
- Expr ∗ **rhs**

    right hand side of the Add expression

5.1.1 Constructor & Destructor Documentation

5.1.1.1 Add()

```
Add::Add (
            Expr * lhs,
            Expr * rhs )
```

Constructor takes two arguments to create Add object.

Parameters

lhs	first argument, Expr∗ type which will be the left hand side of the expression
rhs	second argument, Expr∗ type which will be the right hand side of the expression

Returns

creates Add type

5.1.2 Member Function Documentation

5.1.2.1 Equals()

```
bool Add::Equals (
            Expr * expr )  [virtual]
```

Compares two Add type expressions.

Parameters

expr	first argument, Expr type to which this Add expression will be compared

Returns

boolean value after comparison

Implements Expr.

5.1.2.2 has_variable()

```
bool Add::has_variable ( )  [virtual]
```

checks if the expression has a variable

Returns

boolean value if the expression has a variable

Implements Expr.

5.1.2.3 interp()

```
int Add::interp ( )  [virtual]
```

Interpret the expression to an integer value.

Returns

integer value after calculation

Implements Expr.

5.1.2.4 subst()

```
Expr * Add::subst (
             std::string x,
             Expr * exp )  [virtual]
```

Substitutes a variable present in the expression with a given expression.

Parameters

x	first argument, the variable that will be substituted
exp	second argument, the expression which will substitute the variable

Returns

new expression after substitution

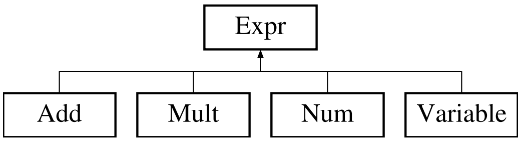
Implements Expr.

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

- /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Add.hpp
- /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Add.cpp

5.2 Expr Class Reference

Inheritance diagram for Expr:



Public Member Functions

- virtual bool Equals (Expr ∗e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr ∗ subst (std::string x, Expr ∗exp)=0

5.2.1 Member Function Documentation

5.2.1.1 Equals()

```
virtual bool Expr::Equals (
             Expr ∗ e )  [pure virtual]
```

Implemented in Add, Mult, Num, and Variable.

5.2.1.2 has_variable()

```
virtual bool Expr::has_variable ( )  [pure virtual]
```

Implemented in Add, Mult, Num, and Variable.

5.2.1.3 interp()

```
virtual int Expr::interp ( )  [pure virtual]
```

Implemented in Add, Mult, Num, and Variable.

5.2.1.4 subst()

```
virtual Expr * Expr::subst (
            std::string x,
            Expr * exp )  [pure virtual]
```

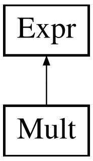
Implemented in Add, Mult, Num, and Variable.

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

- /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Expr.hpp

5.3 Mult Class Reference

Inheritance diagram for Mult:



Public Member Functions

- Mult (Expr *lhs, Expr *rhs)

    Constructor takes two arguments to create Mult object.
- bool Equals (Expr *expr)

    Compares two Mult type expressions.
- int interp ()

    Interpret the expression to an integer value.
- bool has_variable ()

    checks if the expression has a variable
- Expr * subst (std::string x, Expr *exp)

    Substitutes a variable present in the expression with a given expression.

- virtual bool Equals (Expr *e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr * subst (std::string x, Expr *exp)=0

Public Attributes

- Expr * **lhs**

    left hand side of the Mult expression
- Expr * **rhs**

    right hand side of the Mult expression

5.3.1 Constructor & Destructor Documentation

5.3.1.1 Mult()

```
Mult::Mult (
            Expr * lhs,
            Expr * rhs )
```

Constructor takes two arguments to create Mult object.

Parameters

lhs	first argument, Expr∗ type which will be the left hand side of the expression
rhs	second argument, Expr∗ type which will be the right hand side of the expression

Returns

creates Mult type

5.3.2 Member Function Documentation

5.3.2.1 Equals()

```
bool Mult::Equals (
            Expr * expr )  [virtual]
```

Compares two Mult type expressions.

Parameters

expr	first argument, Expr type to which this Mult expression will be compared

Returns

boolean value after comparison

Implements Expr.

5.3.2.2 has_variable()

```
bool Mult::has_variable ( )  [virtual]
```

checks if the expression has a variable

Returns

boolean value if the expression has a variable

Implements Expr.

5.3.2.3 interp()

```
int Mult::interp ( )  [virtual]
```

Interpret the expression to an integer value.

Returns

integer value after calculation

Implements Expr.

5.3.2.4 subst()

```
Expr * Mult::subst (
            std::string x,
            Expr * exp )  [virtual]
```

Substitutes a variable present in the expression with a given expression.

Parameters

x	first argument, the variable that will be substituted
exp	second argument, the expression which will substitute the variable

Returns

new expression after substitution

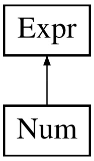
Implements Expr.

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

- /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Mult.hpp
- /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Mult.cpp

5.4 Num Class Reference

Inheritance diagram for Num:



Public Member Functions

- Num (int val)

  Constructor takes one argument to create Num object.
- bool Equals (Expr ∗expr)

  Compares two Num type expressions.
- int interp ()

  Interpret the expression to an integer value.
- bool has_variable ()

  checks if the expression has a variable
- Expr ∗ subst (std::string x, Expr ∗exp)

  Substitutes a variable present in the expression with a given expression, in this no substitution.

- virtual bool Equals (Expr ∗e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr ∗ subst (std::string x, Expr ∗exp)=0

Public Attributes

- int **val**

  integer value that represent the value of the Num expression

5.4.1 Constructor & Destructor Documentation

5.4.1.1 Num()

```
Num::Num (
            int val )
```

Constructor takes one argument to create Num object.

Parameters

int	first argument, int type which will be set as the value for val.

Returns

creates Add type

5.4.2 Member Function Documentation

5.4.2.1 Equals()

```
bool Num::Equals (
            Expr * expr )  [virtual]
```

Compares two Num type expressions.

Parameters

expr	first argument, Expr type to which this Num expression will be compared

Returns

boolean value after comparison

Implements Expr.

5.4.2.2 has_variable()

```
bool Num::has_variable ( )  [virtual]
```

checks if the expression has a variable

Returns

always returns boolean value as the expression is an integer

Implements Expr.

5.4.2.3 interp()

```
int Num::interp ( )  [virtual]
```

Interpret the expression to an integer value.

Returns

integer value

Implements Expr.

5.4.2.4 subst()

```
Expr * Num::subst (
            std::string x,
            Expr * exp )  [virtual]
```

Substitutes a variable present in the expression with a given expression, in this no substitution.

Parameters

x	first argument, the variable that will be substituted
exp	second argument, the expression which will substitute the variable

Returns

new expression with same integer value

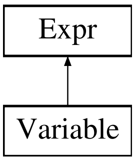
Implements Expr.

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

- /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Num.hpp
- /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Num.cpp

5.5 Variable Class Reference

Inheritance diagram for Variable:



Public Member Functions

- Variable (std::string var)

  Constructor takes one argument to create Mult object.
- bool Equals (Expr ∗expr)

  Compares two Variable type expressions.
- int interp ()

  Interpret the expression to an integer value.
- bool has_variable ()

  checks if the expression has a variable
- Expr ∗ subst (std::string x, Expr ∗exp)

  Substitutes a variable present in the expression with a given expression.

- virtual bool Equals (Expr ∗e)=0
- virtual int interp ()=0
- virtual bool has_variable ()=0
- virtual Expr ∗ subst (std::string x, Expr ∗exp)=0

Public Attributes

- std::string **var**

  variable which represents the expression of the Variable class

5.5.1 Constructor & Destructor Documentation

5.5.1.1 Variable()

```
Variable::Variable (
            std::string var )
```

Constructor takes one argument to create Mult object.

Parameters

var	first argument, string type which will be the argument

Returns

Variable type

5.5.2 Member Function Documentation

5.5.2.1 Equals()

```
bool Variable::Equals (
            Expr * expr )  [virtual]
```

Compares two Variable type expressions.

Parameters

expr	first argument, Expr type to which this Variable expression will be compared

Returns

boolean value after comparison

Implements Expr.

5.5.2.2 has_variable()

```
bool Variable::has_variable ( )  [virtual]
```

checks if the expression has a variable

Returns

boolean value if the expression has a variable, always true in this case

Implements Expr.

5.5.2.3 interp()

```
int Variable::interp ( )  [virtual]
```

Interpret the expression to an integer value.

Returns

integer value after calculation, in this case error

Implements Expr.

5.5.2.4 subst()

```
Expr * Variable::subst (
            std::string x,
            Expr * exp )  [virtual]
```

Substitutes a variable present in the expression with a given expression.

Parameters

x	first argument, the variable that will be substituted
exp	second argument, the expression which will substitute the variable

Returns

new expression after substitution

Implements Expr.

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

- /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Variable.hpp
- /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Variable.cpp

Chapter 6

File Documentation

6.1 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/↩ Expression/Add.cpp File Reference

contains Add class definition It takes two parameters of Expr type as an input. It can be used for getting the result of the sum of the two input expression

```
#include "Add.hpp"
#include "Variable.hpp"
#include "Num.hpp"
```

6.1.1 Detailed Description

contains Add class definition It takes two parameters of Expr type as an input. It can be used for getting the result of the sum of the two input expression

Author

Author Name

6.2 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/↩ Expression/Add.hpp File Reference

Add class.

```
#include <stdio.h>
#include "Expr.hpp"
```

Classes

- class Add

6.2.1 Detailed Description

Add class.

Holds the signature of all the methods present in the Add class

6.3 Add.hpp

Go to the documentation of this file.

```
//
//  Add.hpp
//
//
//  Created by Avishek Choudhury on 1/24/23.
//

#ifndef Add_hpp
#define Add_hpp

#include <stdio.h>
#include "Expr.hpp"

class Add : public Expr {
public:
    Expr* lhs;
    Expr* rhs;

    Add(Expr* lhs, Expr* rhs);
    bool Equals(Expr* expr);
    int interp();
    bool has_variable();
    Expr* subst(std::string x, Expr* exp);
};

#endif /* Add_hpp */
```

6.4 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/cmdline.cpp File Reference

contains cmdline class definition It tis responsible for analysing the input given by the user and perform the next set of actions

```
#include "catch.h"
#include <iostream>
#include "cmdline.hpp"
```

Functions

- void use_arguments (int argc, char **argv)

  Checks for the command line user input to perform the next steps.

6.4.1 Detailed Description

contains cmdline class definition It tis responsible for analysing the input given by the user and perform the next set of actions

Author

Author Name

6.4.2 Function Documentation

6.4.2.1 use_arguments()

```
void use_arguments (
             int argc,
             char ** argv )
```

Checks for the command line user input to perform the next steps.

Parameters

argc	first argument -> argument count in the input
argv	second argument -> array of the arguments

Returns

void

6.5 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/↩ Expression/cmdline.hpp File Reference

Add class.

Functions

- void use_arguments (int argc, char ∗∗argv)
  Checks for the command line user input to perform the next steps.

6.5.1 Detailed Description

Add class.

Holds the signature for the use_arguments function

6.5.2 Function Documentation

6.5.2.1 use_arguments()

```
void use_arguments (
             int argc,
             char ** argv )
```

Checks for the command line user input to perform the next steps.

Parameters

argc	first argument -> argument count in the input
argv	second argument -> array of the arguments

Returns

void

6.6 cmdline.hpp

Go to the documentation of this file.

```

void use_arguments(int argc, char **argv);
```

6.7 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Expr.hpp File Reference

Expr class.

```
#include "catch.h"
#include <stdio.h>
```

Classes

- class Expr

6.7.1 Detailed Description

Expr class.

Holds the signature for all the methods of the Expr class

Holds the signature for all the methods of the Mult class

Holds the signature for all the methods of the Num class

6.8 Expr.hpp

Go to the documentation of this file.

```
//
//  Expr.hpp
//
//
//  Created by Avishek Choudhury on 1/24/23.
//

#ifndef Expr_hpp
#define Expr_hpp

#include "catch.h"
#include <stdio.h>

class Expr {
public:
    virtual bool Equals(Expr* e) = 0;
    virtual int interp() = 0;
    virtual bool has_variable() = 0;
    virtual Expr* subst(std::string x, Expr* exp) = 0;
};

#endif /* Expr_hpp */
```

6.9 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Mult.cpp File Reference

contains Mult class definition It takes two parameters of Expr type as an input. It can be used for getting the result of the multiplication of the two input expression

```
#include "Mult.hpp"
#include "Variable.hpp"
#include "Num.hpp"
```

6.9.1 Detailed Description

contains Mult class definition It takes two parameters of Expr type as an input. It can be used for getting the result of the multiplication of the two input expression

Author

Avishek

6.10 Mult.hpp

```
//
//  Mult.hpp
//
//
//  Created by Avishek Choudhury on 1/24/23.
//

#ifndef Mult_hpp
#define Mult_hpp

#include <stdio.h>
#include "Expr.hpp"

class Mult : public Expr {
public:
    Expr* lhs;
    Expr* rhs;

    Mult(Expr* lhs, Expr* rhs);
    bool Equals(Expr* expr);
    int interp();
    bool has_variable();
    Expr* subst(std::string x, Expr* exp);
};

#endif /* Mult_hpp */
```

6.11 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/Expression/Num.cpp File Reference

contains Num class definition It takes one parameter of type int as an input. It can be used for creating an object of type Num which has the integer value.

```
#include "Num.hpp"
#include "Variable.hpp"
```

6.11.1 Detailed Description

contains Num class definition It takes one parameter of type int as an input. It can be used for creating an object of type Num which has the integer value.

Author

Avishek

6.12 Num.hpp

```
//
//  Num.hpp
//
//
//  Created by Avishek Choudhury on 1/24/23.
//

#ifndef Num_hpp
#define Num_hpp

#include <stdio.h>
#include "Expr.hpp"

class Num : public Expr {
public:
    int val;

    Num(int val);
    bool Equals(Expr* expr);
    int interp();
    bool has_variable();
    Expr* subst(std::string x, Expr* exp);
};

#endif /* Num_hpp */
```

6.13 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/↩ Expression/Test.cpp File Reference

contains Test class definition It performs testing of all the classes associated with MSDScript

```
#include "catch.h"
#include "Mult.hpp"
#include "Add.hpp"
#include "Num.hpp"
#include "Variable.hpp"
#include <iostream>
```

Functions

- **TEST_CASE** ("All Tests")

6.13.1 Detailed Description

contains Test class definition It performs testing of all the classes associated with MSDScript

Author

Avishek

6.14 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/↩ Expression/Variable.cpp File Reference

contains Variable class definition It takes one parameter of type string as an input. It can be used for creating an object of type Variable which stores string as it's value.

```
#include "Variable.hpp"
```

6.14.1 Detailed Description

contains Variable class definition It takes one parameter of type string as an input. It can be used for creating an object of type Variable which stores string as it's value.

Author

Avishek

6.15 /Users/avishekchoudhury/MSD-CS6015/MSD-CS6015/Expression/↩ Expression/Variable.hpp File Reference

Variable class.

```
#include <stdio.h>
#include <string>
#include "Expr.hpp"
```

Classes

- class Variable

6.15.1 Detailed Description

Variable class.

Holds the signature for all the methods of the Variable class

6.16 Variable.hpp

Go to the documentation of this file.

```
//
//  Variable.hpp
//
//
//  Created by Avishek Choudhury on 1/24/23.
//

#ifndef Variable_hpp
#define Variable_hpp

#include <stdio.h>
#include <string>
#include "Expr.hpp"

class Variable : public Expr {
public:
    std::string var;

    Variable(std::string var);
    bool Equals(Expr* expr);
    int interp();
    bool has_variable();
    Expr* subst(std::string x, Expr* exp);
};

#endif /* Variable_hpp */
```

Index