**Reviewer Name**: Oliver Wang

**Reviewed Name**: Jie-Si, Yang

**Code coverage analysis**:

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Code coverage** | **Proposed test(s) to include** |
| NumExpr::equals(Expr \* e) | 77% |  |
| VarExpr::equals(Expr \* e) | 71.4% |  |
| AddExpr::equals(Expr \* e) | 71.4% |  |
| MultExpr::equals(Expr \* e) | 71.4% |  |
| NumExpr::interp() | 100% |  |
| VarExpr::interp() | 100% |  |
| AddExpr::interp() | 100% |  |
| MultExpr::interp() | 100% |  |
| NumExpr::has\_variable() | 100% |  |
| VarExpr::has\_variable() | 100% |  |
| AddExpr::has\_variable() | 100% |  |
| MultExpr::has\_variable() | 100% |  |
| NumExpr::subst(std::string, Expr\*e) | 100% |  |
| VarExpr::subst(std::string, Expr\*e) | 83.3% | Include test for subst() of single VarExpr |
| AddExpr::subst(std::string, Expr\*e) | 100% |  |
| MultExpr::subst(std::string, Expr\*e) | 0% | Include some tests for subst() of Multexpr |

**Thoughts / suggestions to improve the code or the tests**:

|  |
| --- |
| Cover more edge cases like 0 or empty string. |
| Test some commutative cases. |
| Test more complex nested expression. |
| Add more tests of equals(). |

Add rows when necessary.