

Completeness

From all the mandatory functions, all have been done.

Correctness

In some cases when there is a div in a pow, and the division is made with a denominator that is zero, instead of a failure it returns something. And when is a pow in a pow, when the power is less than zero sometime instead of a failure it returns something. Beside that, all other functions work fine without any known issues. As for how to potentially address those problems, we can study more those particular cases, and create an if else statement for those situations.

The correctness of the code was mainly tested with the online tool (<https://find.incorrectness.dk/>), and in many cases the tests were passed. Out of 189 tests we failed only 7, and from them 3 were for the optional part and 4 are because of the pow. Out of the 7 failed test, only 4 failed for implemented functions. And those problems were stated. The cases when it failed for an implemented function are:

- Pow (Div (Cst 4) (Cst 0)) (Cst 0) for evalFull and evalSimple
- Pow (Pow (Cst 2) (Cst (-1))) (Cst 0) for evalFull and evalSimple

But we also created a small test suite to test other cases.

Efficiency

From the point of view of efficiency, we consider that we have a fairly efficient code, although we do not know exactly how to measure these parameters.

Maintainability

The code is in good shape, although there are sequences of code that seems to be copy-pasted segments, as far as we understood they are ok as being the syntax for the functions. As for the test suits, most of the cases are copy pasted segments with changes. From the point of view of comments, the code contains enough comments of important code sequences or helper functions.

Other

In “evalSimple” for the error when power was negative, we created a custom error called “LessThanZeroException” and we created a second handler to catch it. First handler created was for ArithException and it was handling exception “DividedByZero”.