

Programming Paradigms - A Haskell package for univariate expressions

Student: Cristina Ioana Simionescu

My solution to this mini project makes use of a custom data type, which is an expression, that can either be the addition, multiplication, division of two expressions, the derivative, sine or cosine of an expression or a polynomial. The polynomial type can be either a variable represented as a string, a polynomial multiplied by a constant, the addition of two polynomials or a polynomial to the power of a non negative integer.

The program takes the user's input, which should be a mathematical expression in the form of a string. This input is lexed and parsed into an intermediate data type, which is then again transformed to the final data type.

Once this is done, the expression can be evaluated and printed to a .tex file.

Here are a few examples of how the program evaluates certain expressions:

1. Addition

$$(x + (5 + 3)) = \tag{1}$$

$$(x + 8) \tag{2}$$

2. Multiplication

$$(x^2 * x^3) = \tag{3}$$

$$x^5 \tag{4}$$

3. Multiplication by a scalar

$$(5 * (x + 1)) = \tag{5}$$

$$((5 * x) + (5 * 1)) = \tag{6}$$

$$((5 * x) + 5) \tag{7}$$

4. Differentiation

$$\frac{d}{dx}(x^4 + \frac{d}{dx}\cos(x)) = \tag{8}$$

$$((4 * x^3) - \cos(x)) \tag{9}$$

5. Example 13 (from requirements)

$$\frac{((1 * ((\sin(x) * \sin(x)) + (\cos(x) * \cos(x)))) * x^3)}{x^2} = \tag{10}$$

$$x \tag{11}$$