

Task 3 Evaluation of an Expression

Adrian Jonsson Sjödin

Spring Term 2023

Introduction

The purpose of this task was to implement a program that evaluates a mathematical expression containing variables, and returns the result given an environment for the variables.

Method

The way to solve this task was quite similar to the earlier derivatives task. Like there we started with defining the different allowed expressions and what form a literal will take.

After this we started with implementing the functions given in the skeleton code, these being the `eval/2` functions. Here, like in the previous tasks, we start with considering the base cases of what should happen when we want to evaluate a literal. For the number we simply return the number, while for the variable we use Elixir's `Map` module to retrieve the value bound to the variable. For the quotient we have to do a bit more. We want to evaluate fractions to the lowest common denominator, as well as not have zero as a divisor. To do this we use a conditional `if else` clause and the `Integer` module's `gcd/2` function. The code for this can be seen in code overview [1](#)

Having done the two base cases for when

Code Overview 1: Evaluation of quotient

```
def eval({:quotient, dividend, divisor}, _environment) do
  if divisor == 0 do :undefined
  else
    gcd = Integer.gcd(dividend, divisor)
    {:quotient, dividend/gcd, divisor/gcd}
  end
end
```

Code

```

def deriv({:exp, u, {:num, n}}, v) do
  {:mul,
    {:mul, {:num, n}, {:exp, u, {:num, n - 1}}},
    deriv(u, v)
  }
end

```

Code in text p_print/1 and p_print/2.

Result

Code with caption and label for referencing

Code Overview 2: Caption here

```

iex(13)> Deriv.test_all()
Expression: ((((((cos(x) + (5*x)) + ((x)^(3) + 4)) + sin(x)) +
sqrt(x)) + ln((x)^(2))) + (4/(((x*3) + 2))^(2)))

Derivative of expression: ((((((((-1*1)*sin(x)) + ((0*x) +
(5*1))) + (((3*(x)^(2))*1) + 0)) + (1*cos(x)) + (1/(2*sqrt(x))))
+ (((2*(x)^(1))*1)/(x)^(2))) + (((4*-2)*(((1*3) + (x*0)) + 0))/
(((x*3) + 2))^(3)))

Simplified: ((((((((-1*sin(x)) + 5) + (3*(x)^(2))) + cos(x)) +
(1/(2*sqrt(x)))) + ((2*x)/(x)^(2))) + (-24/(((x*3) + 2))^(3)))
:ok

```

Discussion

Link [GitHub](#).